

# **BIOS and Kernel Developer's Guide (BKDG) For AMD Family 10h Processors**

© 2005–2010 Advanced Micro Devices, Inc. All rights reserved.

The contents of this document are provided in connection with Advanced Micro Devices, Inc. ("AMD") products. AMD makes no representations or warranties with respect to the accuracy or completeness of the contents of this publication and reserves the right to make changes to specifications and product descriptions at any time without notice. No license, whether express, implied, arising by estoppel or otherwise, to any intellectual property rights is granted by this publication. Except as set forth in AMD's Standard Terms and Conditions of Sale, AMD assumes no liability whatsoever, and disclaims any express or implied warranty, relating to its products including, but not limited to, the implied warranty of merchantability, fitness for a particular purpose, or infringement of any intellectual property right. AMD's products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or in any other application in which the failure of AMD's product could create a situation where personal injury, death, or severe property or environmental damage may occur. AMD reserves the right to discontinue or make changes to its products at any time without notice.

#### Trademarks

AMD, the AMD Arrow logo, 3DNow!, and combinations thereof are trademarks of Advanced Micro Devices, Inc.

MMX is a trademark of Intel Corporation.

Microsoft is a registered trademark of Microsoft Corporation.

HyperTransport is a licensed trademark of the HyperTransport Technology Consortium.

Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

# Table of Contents

<b>1</b>	<b>Overview</b>	<b>24</b>
1.1	Intended Audience	24
1.2	Reference Documents	24
1.3	Conventions	24
1.3.1	Numbering	24
1.3.2	Arithmetic And Logical Operators	25
1.4	Definitions	25
1.5	Changes Between Revisions and Product Variations	29
1.5.1	Revision Conventions	29
1.5.2	Major Changes Relative to Family 0Fh Processors	30
1.5.3	Changes For Revision C	31
1.5.3.1	Changes for Revision DA-C	32
1.5.3.2	Changes for Revision C3	32
1.5.4	Changes For Revision D	33
1.5.4.1	Changes for Revision D1	34
1.5.4.1.1	Multi-node Processors	34
1.5.5	Changes for Revision E	34
1.5.6	Supported Feature Variations	36
<b>2</b>	<b>Functional Description</b>	<b>38</b>
2.1	Processor Overview	38
2.2	System Overview	39
2.2.1	Mixed Stepping	39
2.3	Processor Initialization	39
2.3.1	BSP initialization	40
2.3.2	AP initialization	40
2.3.3	Using L2 Cache as General Storage During Boot	40
2.3.4	Multiprocessing Capability Detection	42
2.3.5	BIOS Requirements For 64-Bit Operation	43
2.3.6	SLIT and SRAT	43
2.3.6.1	SLIT	43
2.3.6.2	SRAT	44
2.4	Power Management	44
2.4.1	Processor Power Planes And Voltage Control	45
2.4.1.1	VID Pins And Interface Selection	46
2.4.1.2	Internal VID Registers	46
2.4.1.3	MinVid and MaxVid Check	46
2.4.1.4	PSI_L	47
2.4.1.5	Alternative Voltage (Altvid)	47
2.4.1.6	VID Encodings	48
2.4.1.6.1	Boot VID Encodings	48
2.4.1.6.2	Parallel VID Interface (PVI) Encodings	48
2.4.1.6.3	Serial VID (SVI) Encodings	49
2.4.1.7	BIOS Requirements for Power Plane Initialization	50
2.4.1.8	Hardware-Initiated Voltage Transitions	50
2.4.1.9	Software-Initiated Voltage Transitions	50
2.4.1.9.1	Software-Initiated NB Voltage Transitions	51

2.4.1.9.2	Software-Initiated CPU Voltage Transitions	51
2.4.1.10	SVI Protocol	51
2.4.2	P-states	51
2.4.2.1	Core P-states	52
2.4.2.1.1	Core Performance Boost (CPB)	52
2.4.2.1.2	Core P-state Naming and Numbering	53
2.4.2.1.2.1	Software P-state Numbering	53
2.4.2.1.2.2	Hardware P-state Numbering	53
2.4.2.1.3	Core P-state Control	54
2.4.2.2	NB P-states	55
2.4.2.2.1	NB P-state Control	55
2.4.2.3	P-state Limits	55
2.4.2.4	P-state Bandwidth Requirements	55
2.4.2.5	P-state Transition Behavior	56
2.4.2.6	BIOS Requirements for P-State Initialization and Transitions	57
2.4.2.7	BIOS Configuration for Dual-plane Only Support	57
2.4.2.8	BIOS Configuration for Asymmetric Boost	58
2.4.2.9	BIOS Northbridge COF and VID Configuration	58
2.4.2.9.1	BIOS NB COF and VID Configuration for SVI and Single-Plane PVI Systems	59
2.4.2.9.2	BIOS NB COF and VID Configuration for Dual-Plane PVI Systems	59
2.4.2.10	Processor-Systemboard Power Removal Check	60
2.4.2.11	Processor-Systemboard Power Delivery Compatibility Check	60
2.4.2.12	Mixed-Frequency and Power P-State Configuration	62
2.4.2.12.1	Mixed Power P-State Configuration Sequence	62
2.4.2.12.2	Mixed Frequency and Power P-State Configuration Rules	63
2.4.2.12.3	Mixed Frequency and Power P-State Configuration Sequence	63
2.4.2.13	ACPI Processor P-State Objects	69
2.4.2.13.1	_PCT (Performance Control)	69
2.4.2.13.2	_PSS (Performance Supported States)	69
2.4.2.13.3	_PPC (Performance Present Capabilities)	70
2.4.2.13.4	_PSD (P-State Dependency)	70
2.4.2.13.5	Fixed ACPI Description Table (FADT) Entries	71
2.4.2.14	XPSS (Microsoft® Extended PSS) Object	71
2.4.2.15	BIOS COF and VID Requirements After Warm Reset	71
2.4.2.15.1	Core P-State Voltage Alignment After Warm Reset	71
2.4.2.15.2	Core Maximum P-State Transition Sequence After Warm Reset	71
2.4.2.15.3	Core Minimum P-State Transition Sequence After Warm Reset	72
2.4.2.15.4	NB COF and VID Transition Sequence After Warm Reset	73
2.4.3	C-states	73
2.4.3.1	C-state Names and Numbers	73
2.4.3.2	C-state Request Interface	74
2.4.3.3	C1 Enhanced State (C1E)	74
2.4.3.3.1	SMI Initiated C1E	74
2.4.3.3.2	BIOS Requirements to Initialize SMI Initiated C1E	74
2.4.3.3.2.1	SMM Handler Requirements for C1E	74
2.4.3.3.3	Hardware Initiated C1E	75
2.4.3.3.3.1	BIOS Requirements to Initialize Hardware Initiated C1E	75
2.4.3.3.4	Message Triggered C1E	75
2.4.3.3.5	BIOS Requirements to Initialize Message Triggered C1E	76

2.4.3.4	Cache Flush On Halt	76
2.4.3.5	C5	77
2.4.3.6	ACPI Processor C-state Objects	77
2.4.3.6.1	_CST	77
2.4.3.6.2	_CRS	77
2.4.4	ACPI Suspend to RAM State (S3)	77
2.4.5	Effective Frequency	79
2.5	Processor State Transition Sequences	80
2.5.1	ACPI Power State Transitions	80
2.6	The Northbridge (NB)	80
2.6.1	Northbridge (NB) Architecture	80
2.6.2	The GART	81
2.6.3	DMA Exclusion Vectors (DEV)	81
2.6.4	Northbridge Routing	81
2.6.4.1	Address Space Routing	81
2.6.4.1.1	DRAM and MMIO Memory Space	81
2.6.4.1.2	IO Space	82
2.6.4.1.3	Configuration Space	82
2.6.4.2	HyperTransport™ Technology Routing	82
2.6.4.2.1	Routing Table Configuration	84
2.6.4.2.2	NodeId Enumeration Requirements for Dual-node Processors	85
2.6.4.2.3	BIOS Requirements for Systems with Mixed Processor Families	85
2.6.4.2.4	Link Traffic Distribution	85
2.6.4.2.4.1	Coherent Link Traffic Distribution	86
2.6.4.2.4.2	Coherent Link Pair Traffic Distribution	86
2.6.4.2.5	Display Refresh And IFCM	86
2.6.4.2.5.1	Recommended Buffer Count Settings for Single Link Processors	87
2.6.4.2.5.2	Recommended Buffer Count Settings for Multi-link Processors	89
2.6.5	The Level 3 Cache (L3)	93
2.6.5.1	Probe Filter	93
2.6.5.1.1	Probe Filter Errors	94
2.6.6	Memory Scrubbers	94
2.6.7	Physical Address Space	95
2.6.8	System Address Map	95
2.7	Links	95
2.7.1	Link Initialization	95
2.7.1.1	Ganging And Unganging	95
2.7.1.2	Ganging Detection And Control	96
2.7.1.3	Link Type Detect	96
2.7.1.3.1	Unused Links	96
2.7.1.4	Legal Topologies	96
2.7.1.5	Link Mapping for Dual-Node Processors in the G34 Package	97
2.7.2	Termination and Compensation	97
2.7.3	Equalization	98
2.7.4	Link Bandwidth Requirements	98
2.7.5	Link Retry	98
2.7.6	Link LDTSTOP_L Disconnect-Reconnect	99
2.7.7	LDTSTOP Requirements	99
2.7.8	Response Ordering	100
2.7.9	Link Testing, BIST, and ILM	100

2.7.10	Miscellaneous Behaviors and Requirements	100
2.8	DRAM Controllers (DCTs)	101
2.8.1	DCT Configuration Registers	103
2.8.2	DDR Pad to Processor Pin Mapping	104
2.8.3	Support For Multiple Unbuffered Logical DIMMs	105
2.8.4	Burst Length	105
2.8.5	Ganged or Unganged Mode Considerations	105
2.8.6	Routing DRAM Requests	106
2.8.7	DRAM Controller Direct Response Mode	109
2.8.8	DRAM Data Burst Mapping	109
2.8.9	DCT/DRAM Initialization	110
2.8.9.1	DDR3 Supply Voltage Configuration	110
2.8.9.2	Phy and Controller Mode Configuration	111
2.8.9.3	Phy compensation initialization	111
2.8.9.4	SPD ROM-Based Configuration	111
2.8.9.5	Non-SPD ROM-Based Configuration	112
2.8.9.5.1	Trdrd (Read to Read Timing)	112
2.8.9.5.2	Twrrw (Write to Write Timing)	113
2.8.9.5.3	Twrrd (Write to Read DIMM Termination Turn-around)	113
2.8.9.5.4	TrwtTO (Read-to-Write Turnaround for Data, DQS Contention)	114
2.8.9.5.5	TrwtWB (Read-to-Write Turnaround for Opportunistic Write Bursting)	114
2.8.9.5.6	FourActWindow (Four Bank Activate Window or tFAW)	114
2.8.9.5.7	DRAM ODT Control	115
2.8.9.5.8	DRAM Address Timing and Output Driver Compensation Control	120
2.8.9.6	DRAM Device and Controller Initialization	133
2.8.9.6.1	Software DDR2 Device Initialization	135
2.8.9.6.2	Software DDR3 Device Initialization	136
2.8.9.6.2.1	Software Control Word Initialization	139
2.8.9.7	Phy Fence programming	140
2.8.9.8	DRAM Channel Frequency Change	140
2.8.9.9	DRAM Training	141
2.8.9.9.1	Write Levelization Training	141
2.8.9.9.1.1	Write Leveling Seed Value	143
2.8.9.9.2	DQS Receiver Enable Training	144
2.8.9.9.3	DQS Position Training	145
2.8.9.9.4	ECC Byte Lane Training	146
2.8.9.9.5	Calculating MaxRdLatency	147
2.8.9.9.5.1	MaxRdLatency Training	148
2.8.9.9.6	Continuous Pattern Generation	149
2.8.9.10	DRAM Phy Power Savings	150
2.8.10	Memory Interleaving Modes	151
2.8.10.1	Chip Select Interleaving	151
2.8.10.2	Node Interleaving	153
2.8.11	Memory Hoisting	154
2.8.11.1	DramHoleOffset Programming	155
2.8.11.2	DctSelBaseOffset Programming	155
2.8.12	On-Line Spare	156
2.8.12.1	On-Line Spare and CS Interleaving	156
2.8.13	DRAM On DIMM Thermal Management and Power Capping	157
2.9	CPU Core	157

2.9.1	Virtual Address Space	158
2.9.2	CPU Cores and Downcoring	158
2.9.3	Access Type Determination	158
2.9.3.1	Memory Access to the Physical Address Space	159
2.9.3.1.1	Determining The Cache Attribute	159
2.9.3.1.2	Determining The Access Destination for CPU Accesses	159
2.9.4	Timers	160
2.9.5	APIC	160
2.9.5.1	ApicId Enumeration Requirements	160
2.10	Thermal Functions	161
2.10.1	The Tctl Temperature Scale	161
2.10.2	Thermal Diode	161
2.10.3	Temperature-Driven Logic	162
2.10.3.1	PROCHOT_L and Hardware Thermal Control (HTC)	162
2.10.3.2	Software Thermal Control (STC)	162
2.10.3.3	THERMTRIP	163
2.11	Configuration Space	163
2.11.1	MMIO Configuration Coding Requirements	164
2.11.2	MMIO Configuration Ordering	164
2.11.3	Processor Configuration Space	164
2.12	RAS and Advanced Server Features	164
2.12.1	Machine Check Architecture	164
2.12.1.1	Machine Check Registers	165
2.12.1.2	Machine Check Errors	166
2.12.1.2.1	Machine Check Error Logging and Reporting	167
2.12.1.2.2	Error Overflow	167
2.12.1.3	MCA Initialization	168
2.12.1.4	Handling Machine Check Exceptions	168
2.12.1.5	Error Diagnosis	170
2.12.1.5.1	Common Diagnosis Information	170
2.12.1.6	Error Thresholding	171
2.12.1.7	Scrub Rate Considerations	172
2.12.1.8	Error Injection and Simulation	173
2.12.2	DRAM Considerations for ECC	173
2.12.2.1	Multi-Beat Interleaving for Unganged Mode	175
2.12.2.2	Software Managed Bad Symbol Identification	176
2.12.2.3	ECC Syndromes	176
2.12.2.3.1	x8 ECC	177
2.12.2.3.2	x4 ECC	181
2.12.3	Sideband Interface (SBI)	183
2.12.3.1	SBI Processor Information	183
2.13	Interrupts	183
2.13.1	Local APIC	183
2.13.1.1	Physical Destination Mode	184
2.13.1.2	Logical Destination Mode	184
2.13.1.3	Interrupt Delivery	184
2.13.1.4	Vectored Interrupt Handling	184
2.13.1.5	Interrupt Masking	185
2.13.1.6	Spurious Interrupts	185
2.13.1.6.1	Spurious Interrupts Caused by Timer Tick Interrupt	185

2.13.1.7	Lowest-Priority Interrupt Arbitration	185
2.13.1.8	Inter-Processor Interrupts	186
2.13.1.9	APIC Timer Operation	186
2.13.1.10	Generalized Local Vector Table	186
2.13.1.11	State at Reset	186
2.13.2	System Management Mode (SMM)	186
2.13.2.1	SMM Overview	187
2.13.2.2	Operating Mode and Default Register Values	187
2.13.2.3	SMI Sources And Delivery	187
2.13.2.4	SMM Initial State	188
2.13.2.5	SMM Save State	188
2.13.2.6	Exceptions and Interrupts in SMM	194
2.13.2.7	The Protected ASeg and TSeg Areas	194
2.13.2.8	SMM Special Cycles	194
2.13.2.9	Locking SMM	195
2.13.2.10	Multiple Unsynchronized SMI Sources	195
2.14	Secure Virtual Machine Mode (SVM)	196
2.14.1	BIOS support for SVM Disable.	196
2.15	CPUID Instruction.	197
2.15.1	Multi-Core Support	197
2.15.2	L3 Cache Support	197
2.16	Performance Monitoring	197
2.16.1	Performance Monitor Counters	197
2.16.2	Instruction Based Sampling (IBS).	197
<b>3</b>	<b>Registers.</b>	<b>199</b>
3.1	Register Descriptions and Mnemonics	199
3.1.1	Northbridge MSRs In Multi-Core Products	200
3.2	IO Space Registers.	200
3.3	Function 0 HyperTransport™ Technology Configuration Registers	201
3.4	Function 1 Address Map Registers	224
3.5	Function 2 DRAM Controller Registers	233
3.6	Function 3 Miscellaneous Control Registers.	285
3.7	Function 4 Link Control Registers	346
3.8	APIC Registers	372
3.9	CPUID Instruction Registers.	382
3.10	MSRs - MSR000_XXXX	392
3.11	MSRs - MSRC000_0xxx.	410
3.12	MSRs - MSRC001_0xxx.	413
3.13	MSRs - MSRC001_1xxx.	433
3.14	Performance Counter Events.	441
3.14.1	Floating Point Events.	441
3.14.2	Load/Store and TLB Events	443
3.14.3	Data Cache Events.	444
3.14.4	L2 Cache and System Interface Events	447
3.14.5	Instruction Cache Events.	450
3.14.6	Execution Unit Events.	452
3.14.7	Memory Controller Events	456
3.14.8	Crossbar Events	461
3.14.9	Link Events	464



3.14.10 L3 Cache Events .....	464
<b>4 Register List.....</b>	<b>468</b>

# List of Figures

Figure 1:	A single-node processor.....	38
Figure 2:	A dual-node processor.....	38
Figure 3:	System diagram.....	39
Figure 4:	Example 8 node system in twisted ladder topology.....	43
Figure 5:	Two processor G34 topology.....	83
Figure 6:	Four processor G34 maximum performance topology.....	83
Figure 7:	Four processor G34 maximum IO topology.....	84
Figure 8:	Sample four-node configuration.....	84
Figure 9:	Dual-node processor link diagram.....	97
Figure 10:	Link DC termination mode.....	98
Figure 11:	DDR3 x8 Registered DIMM.....	147
Figure 12:	DDR3 Registered x4 DIMM (planar).....	147
Figure 13:	Example cases for programming DramHoleOffset.....	155
Figure 14:	Example cases for programming DctSelBaseOffset.....	156
Figure 15:	Example of line interleaving from x4 DRAM in ungangned DRAM mode using x4 ECC.....	175
Figure 16:	Example of line interleaving from x4 DRAM in ungangned DRAM mode using x8 ECC.....	176
Figure 17:	Address/Command Timing at the Processor Pins.....	258
Figure 18:	Link phy recovered clock and sample clock.....	351

# List of Tables

Table 1:	Processor revision conventions.....	29
Table 2:	Supported feature variations by revision .....	36
Table 3:	SLIT table example without probe filter.....	43
Table 4:	SLIT table example with probe filter.....	44
Table 5:	Power management support.....	44
Table 6:	Boot VID codes.....	48
Table 7:	PVI VID codes.....	48
Table 8:	SVI and internal VID codes.....	49
Table 9:	Voltage transition behavior .....	50
Table 10:	P-state name to MSR address.....	53
Table 11:	P-state name to P-state index .....	54
Table 12:	Representative mixed frequency P-state table example (step 2).....	64
Table 13:	Representative mixed frequency P-state table example (step 3).....	64
Table 14:	Representative mixed frequency P-state table example (step 4).....	65
Table 15:	Representative mixed frequency P-state table example (step 6).....	66
Table 16:	Representative mixed frequency P-state table example (step 7).....	67
Table 17:	Representative mixed frequency P-state table example (final).....	68
Table 18:	Representative mixed frequency _PSS object example.....	68
Table 19:	<a href="#">F0x[F0, D0, B0, 90]</a> and <a href="#">F0x[F4, D4, B4, 94]</a> Recommended Settings.....	87
Table 20:	<a href="#">F3x6C</a> and <a href="#">F3x70</a> Recommended Settings.....	87
Table 21:	.....	88
Table 22:	<a href="#">F2x118</a> , <a href="#">F3x74</a> , and <a href="#">F3x7C</a> Recommended Settings.....	88
Table 23:	<a href="#">F3x140</a> , <a href="#">F3x144</a> , <a href="#">F3x1[54, 50, 4C, 48]</a> , and <a href="#">F3x158</a> Recommended Settings.....	89
Table 24:	<a href="#">F0x[F0, D0, B0, 90]</a> and <a href="#">F0x[F4, D4, B4, 94]</a> Recommended Settings.....	89
Table 25:	<a href="#">F3x6C</a> , <a href="#">F3x70</a> , <a href="#">F3x74</a> , and <a href="#">F3x7C</a> Recommended Settings .....	90
Table 26:	<a href="#">F3x140</a> , <a href="#">F3x144</a> , and <a href="#">F3x1[54, 50, 4C, 48]</a> Recommended Settings for Socket F(1207) and C32.....	91
Table 27:	<a href="#">F3x140</a> and <a href="#">F3x144</a> Recommended Settings for Socket G34.....	92
Table 28:	<a href="#">F3x1[54, 50, 4C, 48]</a> Recommended Settings for 1 Processor G34 System .....	92
Table 29:	<a href="#">F3x1[54, 50, 4C, 48]</a> Recommended Settings for 2 Processor G34 System .....	93
Table 30:	<a href="#">F3x1[54, 50, 4C, 48]</a> Recommended Settings for 4 Processor G34 System .....	93
Table 31:	Link disconnect controls .....	99
Table 32:	Supported link operational modes.....	101
Table 33:	DDR2 Unbuffered and Registered Maximum Frequency Support (per channel).....	102
Table 34:	DDR3 SO-DIMM and Unbuffered Maximum Frequency Support (per channel).....	103
Table 35:	DDR3 Registered Maximum Frequency Support (per channel).....	103
Table 36:	Package pin mapping .....	104
Table 37:	DDR2 Four Bank Activate Window Values .....	115
Table 38:	ODT Settings for DDR2 SO-DIMMs .....	115
Table 39:	ODT Settings for DDR2 UDIMMs and RDIMMs .....	115
Table 40:	ODT Settings for DDR3 SO-DIMMs (1 DIMM per channel) .....	116

Table 41:	ODT Setting for DDR3 SO-DIMMs (2 DIMMs per channel) .....	116
Table 42:	ODT Settings for DDR3 UDIMMs (2 or 3 DIMMs per channel) .....	116
Table 43:	ODT Settings for DDR3 RDIMMs (2 DIMMs per channel).....	116
Table 44:	ODT Settings for DDR3 RDIMMs (3 DIMMs per channel).....	117
Table 45:	DDR3 UDIMM ODT Pattern .....	119
Table 46:	DDR3 SO-DIMM and UDIMM Write Leveling ODT Pattern.....	119
Table 47:	DDR3 RDIMM ODT Pattern (2 DIMMs per channel).....	119
Table 48:	DDR3 RDIMM Write Leveling ODT Pattern (2 DIMMs per channel) .....	119
Table 49:	DDR3 UDIMM and RDIMM ODT Pattern (3 DIMMs per channel).....	120
Table 50:	DDR3 UDIMM and RDIMM Write Leveling ODT Pattern (3 DIMM per channel).....	120
Table 51:	DDR2 SO-DIMM Address Timings and Drive Strengths for 1 DIMM per channel.....	121
Table 52:	DDR2 SO-DIMM Address Timings and Drive Strengths for 2 DIMMs per channel .....	121
Table 53:	DDR2 UDIMM Address Timings and Drive Strengths for 2 DIMMs per channel.....	122
Table 54:	DDR2 RDIMM Address Timings and Drive Strengths for 4 DIMMs per channel .....	123
Table 55:	DDR2 RDIMM Address Timings and Drive Strengths for 2 DIMMs per channel.....	124
Table 56:	AM3 DDR3 UDIMM Address Timings and Drive Strengths .....	124
Table 57:	G34 DDR3 UDIMM Address Timings and Drive Strengths.....	125
Table 58:	C32 DDR3 UDIMM Address Timings and Drive Strengths .....	127
Table 59:	DDR3 SO-DIMM Address Timings and Drive Strengths for 1 DIMM per channel.....	128
Table 60:	DDR3 SO-DIMM Address Timings and Drive Strengths for 2 DIMMs per channel .....	128
Table 61:	DDR3 RDIMM Address Timings and Drive Strengths for 1 DIMM per channel .....	129
Table 62:	G34 DDR3 RDIMM Address Timings and Drive Strengths for 2 DIMMs per channel .....	130
Table 63:	C32 DDR3 RDIMM Address Timings and Drive Strengths for 2 DIMMs per channel .....	130
Table 64:	G34 DDR3 RDIMM Address Timings and Drive Strengths for 3 DIMMs per channel .....	132
Table 65:	C32 DDR3 RDIMM Address Timings and Drive Strengths for 3 DIMMs per channel .....	133
Table 66:	DDR3 RDIMM Register Control Word Values .....	139
Table 67:	Write Leveling Seed Values .....	143
Table 68:	DDR2 swapped normalized address lines for interleaving for a 64-bit interface.....	151
Table 69:	DDR2 swapped normalized address lines for CS interleaving for a 128-bit interface .....	152
Table 70:	DDR3 swapped normalized address lines for interleaving for a 64-bit interface.....	152
Table 71:	DDR3 swapped normalized address lines for CS interleaving for a 128-bit interface .....	153
Table 72:	MCA register cross-reference table .....	165
Table 73:	MC0 and MC4 Overwrite Priorities.....	167
Table 74:	MC1, MC2, MC3, and MC5 Overwrite Priorities .....	168
Table 75:	MCA Registers Commonly Used for Diagnosis.....	170
Table 76:	Recommended Scrub Rates per Node.....	172
Table 77:	Configuration Specific Recommended ECC Symbol Size .....	174
Table 78:	x8 ECC correctable syndromes.....	177
Table 79:	x4 ECC correctable syndromes.....	182
Table 80:	SMM initial state.....	188
Table 81:	SMM Save State.....	189
Table 82:	Terminology in register descriptions.....	199
Table 83:	DIMM support per package .....	234

Table 84:	Logical DIMM, Chip Select, CKE, ODT, and Register Mapping .....	234
Table 85:	DDR2 DRAM address mapping .....	239
Table 86:	DDR3 DRAM address mapping .....	239
Table 87:	Error codes: transaction type.....	292
Table 88:	Error codes: cache level .....	292
Table 89:	Error codes: memory transaction type .....	293
Table 90:	Error codes: participation processor .....	293
Table 91:	Error codes: memory or IO .....	293
Table 92:	NB error descriptions .....	294
Table 93:	NB error signatures, part 1 .....	296
Table 94:	NB error signatures, part 2.....	297
Table 95:	Default MCA NB Address Register default encoding.....	300
Table 96:	MCA NB Address Low Register encoding Protocol Errors .....	300
Table 97:	MCA NB Address Low Register encoding for NB Array Errors .....	302
Table 98:	MCA NB Address Register encoding for L3 Array Errors .....	303
Table 99:	MCA NB Address Low Register encoding for Watchdog Timer Errors .....	303
Table 100:	ACPI Power State Control Register SMAF Settings .....	312
Table 101:	Gen3 deemphasis settings .....	354
Table 102:	Estimated deemphasis (dB) for microstrip routing .....	355
Table 103:	Estimated deemphasis (dB) for stripline routing .....	356
Table 104:	Recommended XmtRdPtr settings for G34 and C32 processors .....	358
Table 105:	Valid ICR field combinations.....	377
Table 106:	DC error descriptions .....	401
Table 107:	DC error signatures .....	402
Table 108:	DC error data; address register.....	402
Table 109:	IC error descriptions.....	404
Table 110:	IC error signatures.....	404
Table 111:	IC error data; address register .....	405
Table 112:	BU error descriptions .....	406
Table 113:	BU error signatures .....	407
Table 114:	BU error data; address register.....	407
Table 115:	LS error signatures .....	408
Table 116:	FR error signatures.....	410

# Revision History

## Revision 3.46 To Revision 3.48 Changes

- Updated Table 2.
- Updated Table 44.
- Updated 2.8.9.5.1 [Trdrd (Read to Read Timing)].
- Clarified F3xE8[CmpCap].
- Clarified MSRC001\_0074[CpuWdtTimeBase].
- Corrected MSRC001\_1037[IbsDcUcMemAcc, IbsDcWcMemAcc].
- Clarified EventSelect 068h.

## Revision 3.34 To Revision 3.46 Changes

- Updated 1.2 [Reference Documents].
- Clarified 2.4.2 [P-states].
- Clarified F0x150[HtRetryCrcCmdPackDynEn].
- Clarified F3xB0[EccErrCntChan].
- Clarified F3x1A0.
- Clarified F3x1EC\_x[FF:01].
- Updated F3x1FC.
- Clarified MSRC000\_04[0F:0B].
- Added EventSelect 068h and EventSelect 069h.
- Updated 2.4.2.11 [Processor-Systemboard Power Delivery Compatibility Check].
- Updated Table 56.
- Updated Table 67.
- Clarified Table 76.
- Clarified F3x74.
- Clarified F3x1[78, 70, 68, 60][Locked].
- Updated 2.4.4 [ACPI Suspend to RAM State (S3)].
- Clarified Table 82.
- Updated F2x[1, 0]90[EnterSelfRef].
- Updated 2.4.2.15.3 [Core Minimum P-State Transition Sequence After Warm Reset].
- Updated 2.4.4 [ACPI Suspend to RAM State (S3)].
- Clarified 2.6.4.2.5 [Display Refresh And IFCM].
- Updated Table 35.
- Corrected Table 36.
- Updated Table 56.
- Added Table 60.
- Updated 2.8.9.9 [DRAM Training].
- Corrected 2.8.9.9.1 [Write Levelization Training].
- Corrected 2.8.9.9.1.1 [Write Leveling Seed Value].
- Added Table 67.
- Clarified F0x164[cHTPrbDistEn,cHTReqDistEn].
- Clarified F0x16C[InLnSt].
- Updated Table 100.
- Clarified CPUID Fn8000\_001B[RdWrOpCnt].
- Updated MSRC001\_0015[TlbCacheDis] and Table 96.
- Clarified MSRC001\_1032[IbsFetchPhysAd] and MSRC001\_1039[IbsDcPhysAd].
- Updated Table 59.

- Updated 2.8.9.9.5 [Calculating MaxRdLatency].
- Clarified Table 77.
- Updated F2x[1, 0]8C[Trfc0].
- Updated F2x9C\_x0A.
- Clarified F2x[1, 0]9C\_x[2B:10].
- Clarified F3x44[ChipKillEccCap].
- Updated 2.4.1.4 [PSI\_L].
- Updated SMMFED8.
- Clarified F1x[BC:80][Lock].
- Updated F4x1[9C, 94, 8C, 84]\_x[D4, C4][DCV].
- Updated APIC410[ExtApicIdEn].
- Updated 2.3.3 [Using L2 Cache as General Storage During Boot].
- Updated 2.4.1.4 [PSI\_L].
- Updated 2.8.9.1 [DDR3 Supply Voltage Configuration].
- Updated Table 84.
- Updated F2x[1, 0]88[MemClkDis].
- Updated F2x[1, 0]9C\_x0C[CKETri, ODTTri, ChipSelTri].
- Added MSRC001\_1021[DIS\_SPEC\_TLB\_RLD, DIS\_IND], MSRC001\_1022[DIS\_SPEC\_TLB\_RLD].
- Added 2.8.2 [DDR Pad to Processor Pin Mapping].
- Corrected F2x9C\_x0A.
- Added F2x[1,0]9C\_x0D0F812F.
- Updated F2x[1,0]9C\_x0D0F2[F:0]30.
- Updated MSRC001\_0116 and MSRC001\_0117.

### Revision 3.28 To Revision 3.34 Changes

- Updated 2.3.3 [Using L2 Cache as General Storage During Boot].
- Updated 2.4.2.5 [P-state Transition Behavior].
- Updated 2.6.5.1 [Probe Filter].
- Updated Table 53.
- Clarified 2.8.9.8 [DRAM Channel Frequency Change].
- Clarified F2x[1, 0]90[DisDllShutdownSR].
- Updated 2.8.9 [DCT/DRAM Initialization].
- Updated Table 46.
- Updated Table 42, Table 49, Table 50, and Table 56.
- Updated Table 62.
- Added 2.8.9.10 [DRAM Phy Power Savings].
- Clarified F2x[1, 0]90[InitDram].
- Updated F2x[1, 0]98[DctAccessDone].
- Added F2x[1,0]9C\_x0D0F[9:8][F:0]0[8,4,0].
- Updated F2x[1, 0]A8[Ddr3FourSocketCh].
- Clarified F3xD8[VSSlamTime].
- Updated 2.4.2.15.2 [Core Maximum P-State Transition Sequence After Warm Reset].
- Updated 2.4.2.15.3 [Core Minimum P-State Transition Sequence After Warm Reset].
- Clarified 2.6.5.1 [Probe Filter].
- Clarified 2.8 [DRAM Controllers (DCTs)].
- Updated 2.8.9.5.1 [Trdrd (Read to Read Timing)], 2.8.9.5.2 [Twrwr (Write to Write Timing)], 2.8.9.5.3 [Twrrd (Write to Read DIMM Termination Turn-around)], 2.8.9.5.4 [TrwtTO (Read-to-Write Turn-around for Data, DQS Contention)].
- Clarified 2.8.10 [Memory Interleaving Modes].
- Updated 2.8.9.9.2 [DQS Receiver Enable Training].

- Updated 2.12.2 [DRAM Considerations for ECC].
- Updated 2.12.2.2 [Software Managed Bad Symbol Identification].
- Updated F2x[1, 0]90[ParEn, PllLockTime].
- Clarified F2x[1, 0]94[PowerDownMode, PowerDownEn].
- Added F2x[1, 0]A8[ActiveCmdAtRst].
- Clarified F3x40.
- Clarified F3x44[CorrMcaExcEn].
- Clarified F3x[84:80][NbGateEn].
- Updated MSRC001\_00[68:64][IddValue].

### Revision 3.20 To Revision 3.28 Changes

- Updated 2.4.3.3.3 [Hardware Initiated C1E].
- Added 1.5.3.1 [Changes for Revision DA-C].
- Added 1.5.4 [Changes For Revision D].
- Updated 2.6.5.1 [Probe Filter].
- Clarified F2x[1, 0]A4[CmdThrottleMode].
- Clarified Table 98.
- Updated F3x1C4.
- Clarified CPUID Fn8000\_0006[L3Size, L3Assoc].
- Updated 2.8.5 [Ganged or Unganged Mode Considerations].
- Updated 2.12.2 [DRAM Considerations for ECC].
- Updated F2x[1, 0]9C\_x0D[RxMaxDurDllNoLock, TxMaxDurDllNoLock].
- Clarified F2x110[MemClrBusy and MemClrInit].
- Clarified F3x44[CorrMcaExcEn].
- Clarified Table 92.
- Clarified 1.5.1 [Revision Conventions].
- Updated 2.4.1.4 [PSI\_L].
- Updated 2.4.2.15.2 [Core Maximum P-State Transition Sequence After Warm Reset].
- Updated 2.4.2.15.3 [Core Minimum P-State Transition Sequence After Warm Reset].
- Clarified 2.6.4.2.5 [Display Refresh And IFCM].
- Clarified Table 94.
- Updated F3xDC[CacheFlushOnHaltTmr].
- Updated 1.4 [Definitions].
- Updated Table 2.
- Updated Table 5.
- Updated 2.4.2.9 [BIOS Northbridge COF and VID Configuration].
- Updated 2.4.2.9.1 [BIOS NB COF and VID Configuration for SVI and Single-Plane PVI Systems].
- Clarified 2.4.4 [ACPI Suspend to RAM State (S3)].
- Updated 2.9.4 [Timers].
- Updated 2.12.1.7 [Scrub Rate Considerations].
- Updated F3x44[NbMcaToMstCpuEn].
- Clarified F3x58.
- Updated F3x64[HtcEn].
- Updated Table 100.
- Updated F3xA0[NbPstateForce].
- Updated F3xD4[NbClkDiv].
- Clarified MSR0000\_0010[TSC].
- Clarified MSR0000\_0417.
- Corrected 2.8.9.7 [Phy Fence programming].
- Clarified 2.16.2 [Instruction Based Sampling (IBS)].



- Clarified F2x[1, 0]7C[EnDramInit], F2x[1, 0]90[InitDram].
- Corrected F2x[1, 0]8C[Twrwr].
- Updated F2x[1, 0]A8[OdtSwizzle].
- Added 2.6.4.2.5.1 [Recommended Buffer Count Settings for Single Link Processors].
- Added 2.6.4.2.5.2 [Recommended Buffer Count Settings for Multi-link Processors].
- Updated 2.8.9.4 [SPD ROM-Based Configuration].
- Clarified 2.8.9.5.1 [Trdrd (Read to Read Timing)], 2.8.9.5.2 [Twrwr (Write to Write Timing)], 2.8.9.5.3 [Twrrd (Write to Read DIMM Termination Turn-around)], 2.8.9.5.4 [TrwtTO (Read-to-Write Turn-around for Data, DQS Contention)].
- Updated 2.8.9.9.5.1 [MaxRdLatency Training].
- Updated F0x[F0, D0, B0, 90].
- Clarified F3x1E8.

### Revision 3.18 To Revision 3.20 Changes

- Updated 2.4.2.13.4 [\_PSD (P-State Dependency)].
- Clarified 2.4.2.15.4 [NB COF and VID Transition Sequence After Warm Reset].
- Updated or added the following sections to include support for AM3 dual-plane only processors:
  - 2.4.2.6 [BIOS Requirements for P-State Initialization and Transitions].
  - 2.4.2.7 [BIOS Configuration for Dual-plane Only Support].
  - 2.4.2.9.1 [BIOS NB COF and VID Configuration for SVI and Single-Plane PVI Systems].
  - 2.4.2.9.2 [BIOS NB COF and VID Configuration for Dual-Plane PVI Systems]
- Clarified 2.4.2.13.2 [\_PSS (Performance Supported States)] and F3xD4[PowerStepUp].
- Updated 2.7.7 [LDTSTOP Requirements].
- Updated Table 32.
- Clarified F0x[F0, D0, B0, 90][LockBc].
- Clarified F0x[14C:130][RetryModeEn] and F0x[18C:170][ScrambleEn].
- Clarified F2x[1, 0]94[ProcOdtDis].
- Clarified F3x58.
- Updated F3x74.
- Updated F3xD8[TdpVid].
- Added F3x1FC[DualPlaneOnly].
- Added EventSelect 1CFh.
- Clarified EventSelect 0F6h.

### Revision 3.06 To Revision 3.18 Changes

- Initial public release for revision C.
- Clarified F2x[1, 0]88[MemClkDis].
- Clarified F2x[1, 0]9C\_x[3:0]0[2:1], F2x[1, 0]9C\_x[3:0]0[6:5], and F2x[1, 0]9C\_x[4A:30].
- Updated F3xA0[PllLockTime].
- Updated 2.7.10 [Miscellaneous Behaviors and Requirements].
- Updated 2.8.9.6.1 [Software DDR2 Device Initialization].
- Updated 2.12.1.5.1 [Common Diagnosis Information].
- Updated F0x[F0, D0, B0, 90], F3x6C, F3x74, and F3x140.
- Clarified F2x[1, 0][5C:40][TestFail].
- Clarified MSR0000\_017B.
- Updated CPUID Fn0000\_0001\_ECX[31].
- Clarified 2.3.1 [BSP initialization] and 2.3.2 [AP initialization].
- Updated 2.4.2.13.4 [\_PSD (P-State Dependency)].
- Updated 2.8.9.5.7 [DRAM ODT Control].

- Corrected 2.8.9.6.1 [Software DDR2 Device Initialization].
- Updated 2.8.9.9.1 [Write Levelization Training].
- Updated 2.8.9.9.2 [DQS Receiver Enable Training].
- Corrected 2.8.9.9.4 [ECC Byte Lane Training].
- Updated 2.12.1.2.1 [Machine Check Error Logging and Reporting].
- Added 2.12.1.3 [MCA Initialization].
- Updated F0x16C[ForceFullT0].
- Clarified F0x18C:170[LS2En].
- Clarified F1xBC:80.
- Updated F2x1, 0]78[MaxRdLatency].
- Clarified F2x1, 0]94[Ddr3Mode].
- Updated F3x5C[ScrubAddrLo] and F3x60[ScrubAddrHi].
- Updated F3x1[78, 70, 68, 60][IntType].
- Updated EventSelect 165h.
- Updated F2x1, 0]90[PendRefPayback].
- Updated F2x1, 0]9C\_x08[WrlvOdt].
- Clarified F3x58.
- Updated Table 53.
- Updated Table 56.
- Updated 2.8.9.6.1 [Software DDR2 Device Initialization].
- Clarified Table 81.
- Clarified F0x150.
- Updated F2x1, 0]98.
- Updated F2x1, 0]9C\_x[3:0]0[6:5].
- Updated MSRC001\_00[68:64][NbDid].
- Clarified MSRC001\_1037[IbsDcMissLat].
- Clarified EventSelect 0CBh.
- Updated 2.4.1.4 [PSI\_L].
- Updated 2.5.1 [ACPI Power State Transitions].
- Updated 2.8.9.5.7 [DRAM ODT Control].
- Updated 2.12.1.4 [Handling Machine Check Exceptions].
- Clarified F3x44[GenCrcErrByte1, GenCrcErrByte0].
- Updated F4x1[9C, 94, 8C, 84]\_x[DF, CF].
- Updated MSR0000\_0400.
- Updated MSRC000\_0081 to MSRC000\_0102 reset values.
- Clarified MSR0000\_0401[CECC, UECC], Table 106, Table 107.
- Clarified 2.8.9.6 [DRAM Device and Controller Initialization].
- Clarified 2.2.1 [Mixed Stepping].
- Updated 2.4.1.4 [PSI\_L].
- Clarified 2.4.3 [C-states].
- Updated 2.5.1 [ACPI Power State Transitions].
- Clarified 2.6.4.2.5 [Display Refresh And IFCM].
- Updated 2.8.9.6.2.1 [Software Control Word Initialization] and Table 66
- Clarified 2.9.3.1.2 [Determining The Access Destination for CPU Accesses].
- Clarified 2.12.1.4 [Handling Machine Check Exceptions].
- Clarified 2.12.2.1 [Multi-Beat Interleaving for Unganged Mode].
- Clarified F0x[F0, D0, B0, 90].
- Clarified F0x150.
- Updated F1x110.
- Corrected F1x114\_x2.
- Clarified F3x1A0[L3FreeListCBC].

- Clarified MSRC001\_0015[TlbCacheDis].
- Clarified MSRC001\_00[49:44].
- Updated 2.7.10 [Miscellaneous Behaviors and Requirements].
- Updated 2.8.9.9.1.1 [Write Leveling Seed Value].
- Corrected 2.8.9.9.3 [DQS Position Training].
- Corrected 2.8.9.9.4 [ECC Byte Lane Training].
- Clarified F2x[1, 0]84[PchgPDMoSel].
- Updated F2x[1, 0]88[Tcl].
- Updated F2x[1, 0]8C[Trdrd, TrwtTO].
- Clarified F2x[1, 0]90[ForceAutoPchg].
- Clarified F2x[1, 0]94[PowerDownMode, FourRankRDimm, DisDramInterface, ZqcsInterval].
- Corrected F2x[1, 0]98[DctAccessWrite].
- Corrected F2x[1, 0]9C\_x04.
- Updated F2x[1, 0]9C\_x08[DisAutoComp].
- Clarified F3xB0[BadDramCs0, SwapEn0].
- Clarified F3xF8\_x4[MceEn].
- Updated 2.3.3 [Using L2 Cache as General Storage During Boot].
- Updated 2.4.2.4 [P-state Bandwidth Requirements].
- Updated 2.7.7 [LDTSTOP Requirements].
- Updated Table 33.
- Corrected Table 55.
- Corrected 2.8.9 [DCT/DRAM Initialization], 2.8.9.9.1 [Write Levelization Training], and 2.8.9.9.1.1 [Write Leveling Seed Value].
- Clarified 2.12.1.7 [Scrub Rate Considerations].
- Updated F0x68[BufRelPri].
- Updated F0x150 recommended values.
- Updated F0x16C[InLnSt].
- Updated F0x[18C:170][ILMEn, BistEn, LS2En].
- Corrected F2x[1, 0]94[PowerDownMode].
- Corrected F2x[1, 0]9C\_x00[ClkDrvStren].
- Updated F4x1[9C, 94, 8C, 84]\_x[D4, C4].
- Clarified MSRC001\_0015[LimitCpuidStdMaxVal].
- Clarified MSRC001\_1036[NbIbsReqDstProc] and MSRC001\_1037[IbsDcMabHit].
- Clarified EventSelect 067h.
- Updated 2.7.1.2 [Ganging Detection And Control].
- Clarified 2.8.8 [DRAM Data Burst Mapping].
- Added 2.13.2.10 [Multiple Unsynchronized SMI Sources].
- Clarified F2x[1, 0]7C[SendMrsCmd].
- Clarified F3xD4[PowerStepUp].
- Corrected APIC[530:500] reset value.
- Clarified MSRC001\_1022[REQ\_CTR].
- Updated 2.4.1 [Processor Power Planes And Voltage Control].
- Clarified 2.7.10 [Miscellaneous Behaviors and Requirements].
- Corrected 2.8.6 [Routing DRAM Requests].
- Clarified 2.8.9.5 [Non-SPD ROM-Based Configuration].
- Clarified 2.8.10.1 [Chip Select Interleaving].
- Clarified F2x[1, 0][5C:40][OnDimmMirror].
- Clarified F2x[1, 0]8C[DisAutoRefresh].
- Clarified F2x[1, 0]94[BankSwizzleMode, ZqcsInterval].
- Updated F2x110[DctSelIntLvAddr].
- Clarified MSRC001\_00[68:64][NbDid].

- Added 2.4.2.10 [Processor-Systemboard Power Removal Check].
- Updated 2.4.2.15.2 [Core Maximum P-State Transition Sequence After Warm Reset].
- Updated 2.4.2.15.3 [Core Minimum P-State Transition Sequence After Warm Reset].
- Clarified 2.4.2.15.4 [NB COF and VID Transition Sequence After Warm Reset].
- Clarified 2.4.4 [ACPI Suspend to RAM State (S3)].
- Clarified 2.14.1 [BIOS support for SVM Disable].
- Updated Table 82.
- Clarified F3x44[IoRdDatErrEn, CpuRdDatErrEn].
- Clarified F3xD8[TdpVid].
- Added MSRC001\_102A[ThrottleNbInterface].
- Updated EventSelect 08Ch.
- Clarified 1.4 [Definitions].
- Clarified 2.8.9.6.2 [Software DDR3 Device Initialization].
- Updated 2.12.1.4 [Handling Machine Check Exceptions].
- Added 2.12.1.5 [Error Diagnosis].
- Clarified 2.12.1.6 [Error Thresholding]
- Updated 2.12.2 [DRAM Considerations for ECC].
- Corrected Figure 15.
- Corrected F1x[1, 0][7C:40][DramBase, IntlvEn] reset values.
- Clarified F2x[1, 0]7C[DeassertMemRstX].
- Clarified F2x[1, 0]94[DisDramInterface].
- Updated F2x[1, 0]9C\_x0C.
- Updated F3x44 and F3x180 recommendations.
- Corrected Table 92 and Table 94 PCC setting for Link Data Error.
- Clarified F3x180[DisPciCfgCpuMstAbtRsp].
- Corrected MSRC001\_00[07:04][CTR].

### Revision 3.00 To Revision 3.06 Changes

- Updated MSRC001\_0070[NbDid].
- Clarified 2.4.4 [ACPI Suspend to RAM State (S3)].
- Clarified 2.8.9.6 [DRAM Device and Controller Initialization].
- Clarified 2.8.9.6.1 [Software DDR2 Device Initialization].
- Clarified 2.8.9.6.2 [Software DDR3 Device Initialization].
- Updated 2.8.9.8 [DRAM Channel Frequency Change].
- Corrected 2.9.3.1.2 [Determining The Access Destination for CPU Accesses].
- Clarified F2x[1, 0]7C[EnDramInit].
- Clarified F2x[1, 0]9C\_x04 programming requirements.
- Clarified F3x44[DisPciCfgCpuErrRsp].
- Clarified F3x180[DisPciCfgCpuMstAbtRsp].
- Clarified 2.4.2.12.2 [Mixed Frequency and Power P-State Configuration Rules].
- Updated 2.4.2.12.3 [Mixed Frequency and Power P-State Configuration Sequence].
- Updated 2.4.2.15.2 [Core Maximum P-State Transition Sequence After Warm Reset].
- Updated 2.4.2.15.3 [Core Minimum P-State Transition Sequence After Warm Reset].
- Clarified 2.6.6 [Memory Scrubbers].
- Added 2.12.1.7 [Scrub Rate Considerations].
- Clarified 2.12.2 [DRAM Considerations for ECC].
- Clarified F3x40[UECC, CECC].
- Clarified F3x4C.
- Clarified F3x58.
- Clarified MSRC001\_0015[McStatusWrEn].

- Added BSC to 1.4 [Definitions].
- Clarified 2.3 [Processor Initialization].
- Clarified 2.4.2.12.2 [Mixed Frequency and Power P-State Configuration Rules].
- Updated 2.4.2.15.3 [Core Minimum P-State Transition Sequence After Warm Reset].
- Updated 2.4.4 [ACPI Suspend to RAM State (S3)].
- Clarified F2x[1, 0]94[PowerDownMode].
- Updated F3xD8[VSSlamTime].
- Clarified F3xF0.
- Updated CPUID Fn8000\_0006[ECX].
- Updated 2.8.9.2 [Phy and Controller Mode Configuration].
- Clarified 2.13.1 [Local APIC].
- Clarified 2.13.1.11 [State at Reset].
- Updated F3x44[WDTBaseSel].
- Added F3x1EC\_x[FF:01].
- Updated F4x1[9C, 94, 8C, 84]\_x[D4, C4]
- Clarified APICF0[APICSWEn].
- Clarified MSR0000\_001B[ApicEn].
- Updated EventSelect 0E0h.
- Added TDP to 1.4 [Definitions].
- Updated Table 2.
- Updated 2.3.3 [Using L2 Cache as General Storage During Boot].
- Updated 2.7.10 [Miscellaneous Behaviors and Requirements].
- Updated 2.8.9.3 [Phy compensation initialization].
- Clarified RDQS in 2.8.9.6.1 [Software DDR2 Device Initialization].
- Clarified TDQS in 2.8.9.6.2 [Software DDR3 Device Initialization].
- Updated 2.8.9.9.2 [DQS Receiver Enable Training].
- Updated 2.8.9.9.3 [DQS Position Training].
- Clarified 2.9.2 [CPU Cores and Downcoring].
- Updated 2.12.1.6 [Error Thresholding]
- Updated 2.12.2 [DRAM Considerations for ECC].
- Updated F0x16C[ForceFullT0] BIOS recommendation.
- Clarified F2x[1, 0]88[Trtp].
- Clarified F2x[1, 0]94[RDqsEn].
- Updated F2x[1, 0]9C\_x[3:0]0[2:1] and F2x[1, 0]9C\_x[3:0]03.
- Updated F2x[1, 0]9C\_x[3:0]0[6:5] and F2x[1, 0]9C\_x[3:0]07.
- Clarified F3xD4.
- Added F3xD8[TdpVid].
- Updated F3x1[78, 70, 68, 60].
- Clarified MSR0000\_0410, MSR0000\_0411, MSR0000\_0412.
- Updated MSR0000\_0413.
- Added MSRC001\_001F[DisXdsBypass].
- Clarified MSRC001\_102A[CILinesToNbDis].
- Added MSRC001\_102A[FrcWTMemTypToWPDIs].
- Updated 1.5.2 [Major Changes Relative to Family 0Fh Processors].
- Updated 2.4.1 [Processor Power Planes And Voltage Control].
- Clarified 2.4.2 [P-states].
- Updated 2.4.2.9.1 [BIOS NB COF and VID Configuration for SVI and Single-Plane PVI Systems].
- Updated 2.4.2.15.2 [Core Maximum P-State Transition Sequence After Warm Reset].
- Updated F0x[E4, C4, A4, 84][LdtStopTriEn].
- Updated F3xE8[MpCap].
- Clarified APIC340 behavior.

- Updated F2x11C[MctPrefReqLimit] and F3x78[RspDBC] requirements.
- Updated MSRC001\_0114 field definitions.
- Updated 1.2 [Reference Documents].
- Updated 2.3.3 [Using L2 Cache as General Storage During Boot].
- Clarified 2.4.2.12.2 [Mixed Frequency and Power P-State Configuration Rules].
- Clarified 2.4.2.13 [ACPI Processor P-State Objects].
- Updated 2.4.4 [ACPI Suspend to RAM State (S3)].
- Clarified F2x11C recommendation to enable prefetcher.
- Updated Table 100 C1E setting.
- Updated MSRC001\_102A[CILinesToNbDis].
- Added IOMMU to 1.4 [Definitions].
- Updated 2.4.2.11 [Processor-Systemboard Power Delivery Compatibility Check].
- Updated Table 42.
- Updated 2.8.9.9.5.1 [MaxRdLatency Training].
- Updated 2.8.12 [On-Line Spare].
- Updated 2.9.3.1.2 [Determining The Access Destination for CPU Accesses] with programming notes for cacheable IO.
- Updated 2.12.1.6 [Error Thresholding].
- Updated Table 82. Added Write-only attribute.
- Added F2x[1, 0]78[ChSetupSync].
- Updated F3x40[ProtEn].
- Updated Table 92 Protocol Error.
- Updated Table 93 and Table 94 with combined link and L3 protocol errors.
- Corrected Table 94: L3 Cache LRU Error/Syndrome Valid.
- Updated Table 96 with contents of table MCA NB Address Low Register encoding for L3 Protocol Errors
- Clarified F3x140 IsocPreqTok requirements for display refresh mode.
- Updated F3x180[SncOnProtEn].
- Added F3x1FC[SinglePlaneNbIdd].
- Updated F4x1[98, 90, 88, 80][LinkPhyOffset] to be 10 bits
- Added MSR0000\_02FF[MtrrDefTypeFixEn] access rule.
- Added NBC, XBAR, and updated WDT in 1.4 [Definitions].
- Added 2.3.6 [SLIT and SRAT].
- Updated 2.6.2 [The GART].
- Updated 2.8.6 [Routing DRAM Requests].
- Added 2.8.7 [DRAM Controller Direct Response Mode].
- Updated 2.8.8 [DRAM Data Burst Mapping].
- Updated 2.8.9.5.1 [Trdrd (Read to Read Timing)].
- Updated 2.8.9.5.2 [Twrwr (Write to Write Timing)].
- Updated 2.8.9.5.3 [Twrrd (Write to Read DIMM Termination Turn-around)].
- Updated 2.8.9.5.4 [TrwtTO (Read-to-Write Turnaround for Data, DQS Contention)].
- Added MC5 to 2.12.1.2.2 [Error Overflow], and to Table 74.
- Added 2.12.1.8 [Error Injection and Simulation].
- Updated 2.13.1.3 [Interrupt Delivery] with sentence from 2.13.1.3 [Interrupt Delivery].
- Clarify effect of F3x44[NbMcaToMstCpuEn] in multi-core processors.
- Updated Table 92 on page 294.
- Updated Table 99 on page 303.
- Updated F4x1[9C, 94, 8C, 84]\_x[D5, C5] description with how to determine the appropriate deemphasis setting.
- Updated APIC[1F0:180][TriggerModeBits] description.

**Revision 3.00**

- Initial public release.



## 1 Overview

The AMD family 10h processor (in this document referred to as *the processor*) is a processing unit that supports x86-based instruction sets. The processor includes (a) independent central processing unit cores (referred to as *cores*), (b) high-speed communication interfaces (referred to as *links*) that may be configured for HyperTransport™ technology (referred to as *IO links*) or for AMD-proprietary inter-processor communication (referred to as *coherent links*), and (c) double-data rate 2 (DDR2) or 3 (DDR3) system memory DRAM interfaces.

AMD family 10h processors are distinguished by the combined ExtFamily and BaseFamily fields of the CPUID instruction (see CPUID Fn[8000\_0001, 0000\_0001]\_EAX in section 3.9 [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 Media Instructions, #26568.
- AMD64 Architecture Programmer's Manual Volume 5: 64-Bit Media and x87 Floating-Point Instructions, #26569.
- CPUID Specification, #25481.
- Software Optimization Guide for AMD Family 10h Processors, #40546.
- Revision Guide for AMD Family 10h Processors, #41322
- AMD I/O Virtualization Technology (IOMMU) Specification, #34434.
- HyperTransport™ I/O Link Specification. [www.hypertransport.org](http://www.hypertransport.org).
- JEDEC standards. [www.jedec.org](http://www.jedec.org)
- PCI local bus specification. [www.pcisig.org](http://www.pcisig.org).
- System Management Bus (SMBus) specification. [www.smbus.org](http://www.smbus.org).

### 1.3 Conventions

#### 1.3.1 Numbering

- **Binary numbers.** Binary numbers are indicated by appending a “b” at the end, e.g., 0110b.
- **Decimal numbers.** Unless specified otherwise, all numbers are decimal. Note: this rule does not apply to the register mnemonics described in section 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 follow some Verilog conventions for logic equations.

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

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

For example, the equation:

$$\text{Output}[3:0] = \{A[1:0], B[3:2]\} \& C[3:0] \mid \sim D[3:0] \& E[9:6],$$

is translated as:

$$\begin{aligned} \text{Output}[3] &= (A[1] \& C[3]) \mid (\sim D[3] \& E[9]); \\ \text{Output}[2] &= (A[0] \& C[2]) \mid (\sim D[2] \& E[8]); \\ \text{Output}[1] &= (B[3] \& C[1]) \mid (\sim D[1] \& E[7]); \\ \text{Output}[0] &= (B[2] \& C[0]) \mid (\sim D[0] \& E[6]); \end{aligned}$$

### 1.4 Definitions

- **AP.** Application processor. See [2.3 \[Processor Initialization\]](#).
- **APML.** Advanced Platform Management Link. See [2.12.3 \[Sideband Interface \(SBI\)\]](#).
- **BCS.** Base configuration space. See [2.11 \[Configuration Space\]](#).
- **BERT.** Bit error rate tester. A piece of test equipment that generates arbitrary test patterns and checks that a device under test returns them without errors.
- **BIST.** Built-in self-test. Hardware within the processor that generates test patterns and verifies that they are stored correctly (in the case of memories) or received without error (in the case of links).
- **Boot VID.** Boot voltage ID. This is the VDD and VDDNB voltage level that the processor requests from the external voltage regulator during the initial phase of the cold boot sequence.
- **BSC.** Boot strap core. Core 0 of the BSP. Specified by [MSR0000\\_001B\[BSC\]](#).
- **BSP.** Boot strap processor. See [2.3 \[Processor Initialization\]](#).
- **C0, C1, C2, and C3.** These are ACPI-defined core power states. C0 is operational. C1 is when the core is in halt. C2 and C3 are stop-grant states. See [2.4 \[Power Management\]](#).
- **C1E.** C1 enhanced state. Power-savings mode that is employed when all cores of a CMP processor are in the halt state. See [\[The Interrupt Pending and CMP-Halt Register\] MSRC001\\_0055](#).
- **Canonical address.** An address in which the state of the most-significant implemented bit is duplicated in all the remaining higher-order bits, up to bit 63.
- **Channel.** See DRAM channel.

- **Channel interleaved mode.** Mode in which DRAM address space is interleaved between DRAM channels. See 2.8.10 [Memory Interleaving Modes].
- **Chipkill ECC.** An error correcting code which can recover from DRAM device failures. See 2.12.2 [DRAM Considerations for ECC].
- **CMP.** Chip multi-processing. Refers to processors that include multiple cores. See 2.1 [Processor Overview].
- **Coherent fabric.** The coherent fabric includes the DRAM controllers and caches of the system. Normally, this refers to the nodes, system memory, and coherent links used for communication between the nodes. See 2.2 [System Overview].
- **Coherent link or coh link.** A link configured for coherent inter-processor traffic between nodes.
- **COF.** Current operating frequency of a given clock domain. See 2.4.2 [P-states].
- **Cold reset.** PWROK is deasserted and RESET\_L is asserted. See 2.3 [Processor Initialization].
- **CPU or CPU core.** The instruction execution unit(s) of the processor. See 2.1 [Processor Overview].
- **CpuCoreNum.** Specifies the core number. See 2.9.2 [CPU Cores and Downcoring].
- **CPUID function X.** Refers to the CPUID instruction when EAX is preloaded with X. See 3.9 [CPUID Instruction Registers].
- **CS.** Chip select. See F2x[1, 0][5C:40] [DRAM CS Base Address Registers].
- **DC coupled.** Refers to the method used for link termination. See 2.7.2 [Termination and Compensation].
- **DCQ.** DRAM controller queue.
- **DCT.** DRAM controller. See 2.8 [DRAM Controllers (DCTs)].
- **DEV.** DMA exclusion vector. See 2.6.3 [DMA Exclusion Vectors (DEV)].
- **DID.** Divisor identifier. Specifies the post-PLL divisor used to reduce the COF. See 2.4.2 [P-states].
- **Display refresh.** Traffic used for display refresh in UMA systems. See 2.6.4.2.5 [Display Refresh And IFCM].
- **Doubleword.** A 32-bit value.
- **Downcoring.** Removal of cores. See 2.9.2 [CPU Cores and Downcoring].
- **DR.** Dual rank DIMM.
- **DRAM channel.** The part of the DRAM interface that connects to a 64-bit DIMM. For example, a processor with a 128-bit DRAM interface is said to support two DRAM channels. See 2.8 [DRAM Controllers (DCTs)].
- **DS.** Downstream. Refers to the direction of data on a link.
- **Dual-Plane.** Refers to a processor or systemboard where VDD and VDDNB are separate and may operate at independent voltage levels. Refer to 2.4.1 [Processor Power Planes And Voltage Control].
- **DW or DWORD.** Doubleword. A 32-bit value.
- **ECS.** Extended configuration space. See 2.11 [Configuration Space].
- **EDS.** Electrical data sheet. See 1.2 [Reference Documents].
- **FDS.** Functional data sheet; there is one FDS for each package type. See 1.2 [Reference Documents].
- **FID.** Frequency identifier. Specifies the PLL frequency multiplier for a given clock domain. See 2.4.2 [P-states].
- **Ganged.** A link, memory channel, or voltage regulator in which all portions are controlled as one.
- **Gen1.** Refers to older revisions of the link specification and, in particular, link data rates from 0.4 to 2.0 GT/s. See 2.7 [Links].
- **Gen3.** Refers to revision 3.00 of the link specification and, in particular, link data rates from 2.4 to 5.2 GT/s. See 2.7 [Links].
- **#GP.** A general-protection exception.
- **#GP(0).** Notation indicating a general-protection exception (#GP) with error code of 0.
- **GT/s.** Giga-transfers per second.
- **HTC.** Hardware thermal control. See 2.10.3.1 [PROCHOT\_L and Hardware Thermal Control (HTC)].
- **HTC-active state.** Hardware-controlled lower-power, lower-performance state used to reduce temperature. See 2.10.3.1 [PROCHOT\_L and Hardware Thermal Control (HTC)].
- **I2C.** Protocol on which the SVI interface timing is based. See 2.4.1 [Processor Power Planes And Voltage

- Control], and section 1.2 [Reference Documents].
- **IBS.** Instruction based sampling. See 2.16.2 [Instruction Based Sampling (IBS)].
  - **IFCM.** Isochronous flow-control mode, as defined in the *HyperTransport™ I/O Link Specification*. See 2.6.4.2.5 [Display Refresh And IFCM].
  - **ILM.** 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 [The Link Extended Control Registers] F0x[18C:170].
  - **Inactive lane.** The inactive lanes of a link are the unused upper lanes when the operating link width is lower than the maximum link width. See the *HyperTransport™ I/O Link Specification*.
  - **IO configuration.** Access to configuration space through IO ports CF8h and CFCh. See 2.11 [Configuration Space].
  - **IO Hub.** This is the platform device that contains the bridge to the system BIOS.
  - **IOMMU.** AMD I/O Virtualization Technology. See the *AMD I/O Virtualization Technology Specification*.
  - **IO link.** A link configured for non-coherent traffic, per the *HyperTransport™ I/O Link Specification*.
  - **IORRs.** IO range registers. See [The IO Range Registers Base (IORR\_BASE[1:0])] MSRC001\_00[18, 16].
  - **Isoc.** Isochronous. Isochronous is defined by the link specification.
  - **L1 caches.** The level 1 caches of the core including the instruction cache and the data cache.
  - **L2 cache.** The level 2 cache of each core.
  - **L3 cache.** The level 3 cache that is shared by each of the cores.
  - **Link.** Generic term that may refer to an IO link or a coherent link.
  - **LINT.** Local interrupt.
  - **Logical DIMM.** Either one 64-bit DIMM or two identical DIMMs in parallel to create a 128-bit interface. See 2.8 [DRAM Controllers (DCTs)].
  - **LVT.** Local vector table. A collection of APIC registers that define interrupts for local events. E.g., [The Extended Interrupt [3:0] Local Vector Table Registers] APIC[530:500].
  - **Master abort.** This is a PCI-defined term that is applied to transactions on other than PCI busses. 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.
  - **MCQ.** Memory controller queue. See 2.6.1 [Northbridge (NB) Architecture].
  - **MCT.** Memory controller. See 2.6.1 [Northbridge (NB) Architecture].
  - **MEMCLK.** Refers to the clock signals, M[B, A][3:0]\_CLK, that are driven from the processor to DDR DIMMs.
  - **MMIO.** 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 [The Memory Mapped IO Base/Limit Registers] F1x[BC:80].
  - **MMIO configuration.** Access to configuration space through memory space. See 2.11 [Configuration Space].
  - **MOF.** Maximum operating frequency of the core(s). Normally this is the core COF in P-state 0. See 2.4.2 [P-states].
  - **MSR.** Model specific register. The CPU includes several MSRs for general configuration and control. See 3.10 [MSRs - MSR0000\_xxxx] for the beginning of the MSR register definitions.
  - **MTRR.** Memory-type range register. The MTRRs specify the type of memory associated with various memory ranges. See MSR0000\_00FE, MSR0000\_02[0F:00], MSR0000\_02[6F:68, 59, 58, 50], and MSR0000\_02FF.
  - **NB.** Northbridge. The transaction routing block of the node. See 2.1 [Processor Overview].
  - **NBC.** Node Base Core. The lowest numbered core in the node.
  - **NCLK.** The main Northbridge clock. The NCLK frequency is the NB COF.
  - **NFCM.** Normal flow control mode. This is the flow control mode when neither Display Refresh nor IFCM are in use. See 2.6.4.2.5 [Display Refresh And IFCM].
  - **Node ID.** The identifier assigned to each node, [The Node ID Register] F0x60[NodeId].

- **Node.** See 2.1 [Processor Overview].
- **Normalized address.** Addresses used by DCTs. See 2.6.1 [Northbridge (NB) Architecture].
- **Octword.** A 128-bit value.
- **ODM.** On-DIMM mirroring. See F2x[1, 0][5C:40][OnDimmMirror].
- **ODT.** On-die termination, which is applied DRAM interface signals.
- **ODTS.** DRAM On-die thermal sensor.
- **Operational frequency.** The frequency at which the processor operates. See 2.4 [Power Management].
- **PDS.** Product data sheet. See 1.2 [Reference Documents].
- **PRBS.** Pseudo-random bit sequence.
- **Probe filter.** See 2.6.5.1 [Probe Filter].
- **Processor.** See 2.1 [Processor Overview].
- **PSI.** Power Status Indicator. See 2.4.1.4 [PSI\_L].
- **P-state.** Performance state. See 2.4 [Power Management].
- **PTE.** Page table entry.
- **PVI.** Parallel VID interface. See 2.4.1 [Processor Power Planes And Voltage Control].
- **QR.** Quad rank DIMM.
- **Quadword.** A 64-bit value.
- **RAS.** Reliability, availability and serviceability (industry term). See 2.12 [RAS and Advanced Server Features].
- **RX.** Receiver.
- **SBI.** Sideband Interface. Also referred to as APML. See 2.12.3 [Sideband Interface (SBI)].
- **Scrubber.** Background memory checking logic. See 2.6.6 [Memory Scrubbers].
- **Shutdown.** 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.
- **Single-Plane.** Refers to a processor or systemboard where VDD and VDDNB are tied together and operate at the same voltage level. Refer to 2.4.1 [Processor Power Planes And Voltage Control].
- **Slam.** Refers to change the voltage to a new value in one step (as opposed to stepping). See 2.4.1.8 [Hardware-Initiated Voltage Transitions].
- **SMAF.** System management action field. This is the code passed from the SMC to the processors in STP-CLK assertion messages. The action taken by the processors in response to this message is specified by [The ACPI Power State Control Registers] F3x[84:80].
- **SMBus.** System management bus. Refers to the protocol on which the serial VID interface (SVI) commands and SBI are based. See 2.4.1 [Processor Power Planes And Voltage Control], 2.12.3 [Sideband Interface (SBI)], and section 1.2 [Reference Documents].
- **SMC.** System management controller. This is the platform device that communicates system management state information to the processor through an IO link, typically the system IO hub.
- **SMI.** System management interrupt. See 2.13.2.1 [SMM Overview].
- **SMM.** System management mode. See 2.13.2 [System Management Mode (SMM)].
- **Southbridge.** Same as IO hub.
- **Speculative event.** A performance monitor event counter that counts all occurrences of the event even if the event occurs during speculative code execution.
- **SR.** Single rank DIMM.
- **STC.** Software thermal control. See 2.10.3.2 [Software Thermal Control (STC)].
- **STC-active state.** Software-controlled lower-power, lower-performance state used to reduce temperature. See 2.10.3.2 [Software Thermal Control (STC)].
- **STC thermal zone.** Temperature range which may be programmed to generate interrupts and special bus cycles. See 2.10.3.2 [Software Thermal Control (STC)].
- **Sublink.** An 8-bit-or-less (CAD) block of link signals of a link; each sublink of a link may connect to a different device. See 2.7 [Links].
- **SVI.** Serial VID interface. See 2.4.1 [Processor Power Planes And Voltage Control].
- **SVM.** secure virtual machine. See 2.14 [Secure Virtual Machine Mode (SVM)].

- **Sync flood.** 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.
- **TCB.** Trace capture buffer.
- **TCC.** Temperature calculation circuit. See 2.10 [Thermal Functions].
- **Tctl.** Processor temperature control value. See 2.10.3 [Temperature-Driven Logic].
- **TDP.** Thermal design power.
- **Thermal diode.** A diode connected to the THERMDA and THERMDC pins used for thermal measurements. See 2.10.2 [Thermal Diode].
- **Token.** A scheduler entry used in various Northbridge queues to track outstanding requests. See [The SRI to XCS Token Count Register] F3x140 on Page 461.
- **TX.** Transmitter.
- **UI.** Unit interval. This is the amount of time equal to one half of a clock cycle.
- **UMA.** Unified memory architecture. This is a type of display device that uses a frame buffer located in main memory.
- **Unganged.** A link, memory channel, or voltage regulator in which portions are controlled separately.
- **US.** Upstream. Refers to the direction of data on a link.
- **VDD.** Main power supply to the processor core logic.
- **VDDNB.** Main power supply to the processor NB logic.
- **VID.** Voltage level identifier. See 2.4.1 [Processor Power Planes And Voltage Control].
- **Virtual CAS.** The clock in which CAS is asserted for the burst, N, plus the burst length (in MEMCLKs), minus 1; so the last clock of virtual CAS = N + BL/2 - 1.
- **VRM.** voltage regulator module.
- **Warm reset.** RESET\_L is asserted only (while PWROK stays high). See 2.3 [Processor Initialization].
- **WDT.** Watchdog timer. A timer that detects activity and triggers an error if a specified period of time expires without the activity. For example, see [The CPU Watchdog Timer Register (CpuWdTmrCfg)] MSRC001\_0074 or the NB watchdog timer in [The MCA NB Control Register] F3x40.
- **XBAR.** Crossbar; command packet switch. See 2.6.1 [Northbridge (NB) Architecture].

## 1.5 Changes Between Revisions and Product Variations

### 1.5.1 Revision Conventions

The processor revision is specified by [The Family, Model, Stepping Identifiers] CPUID Fn[8000\_0001, 0000\_0001]\_EAX. This document uses a revision letter or two letter mnemonic followed by revision letter instead of specific model numbers. The following table shows the relationship between revision and model. 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 AMD Family 10h Processors* for additional information about revision determination.

**Table 1: Processor revision conventions**

Revision	CPUID Fn[8000_0001, 0000_0001]_EAX[Model]
B	2h
C	4h, 5h, and 6h
RB-C	4h
BL-C	5h
DA-C	6h
D	8h and 9h
E	Ah

## 1.5.2 Major Changes Relative to Family 0Fh Processors

- CPU core additions:
  - Support for up to 6 cores per node in product variations.
  - High-performance (128-bit internal data path) floating point unit (per core) in product variations.
  - SSE4A instructions.
  - Advanced bit manipulation (ABM) instructions.
  - MWAIT and MONITOR instructions.
  - Misaligned SSE mode.
  - Power management state invariant time stamp counter (TSC).
  - Number of extended LVT interrupts in APIC increased to 4.
  - Increase in the number of large TLB page entries.
  - 1 Gbyte large paging supported.
  - Physical address space increased to 48 bits.
  - All local sources of SMIs (including sources from the cores and from the NB) are broadcast to all cores of all nodes in the system.
- Memory controller (MCT) and DRAM controllers (DCTs) additions:
  - Support for DDR2/DDR3 DIMMs in product variations.
  - DIMMs controlled through two independent DRAM controllers.
  - Write burst and DRAM prefetching performance improvements.
  - Read and write burst support, as required for DRAM training.
- Links and IO additions:
  - HyperTransport™ 3.00 Technology, including support for DC-coupled mode.
  - Link unganging support.
  - Link-defined extended address capability to support up to 64-bit IO addresses.
  - Link-defined UnitID clumping.
  - Link-defined error retry.
  - Link-defined isochronous flow control mode.
  - Link-defined INTx support.
  - Support for independent ordering between requests with different non-zero SeqID values.
  - Support for 3.2 GHz link frequency for revision D and later.
- RAS-related additions:
  - Core disable capability.
  - New error thresholding support for errors associated with links, and the L3 cache; all the thresholding registers are additionally accessible through configuration space.
  - New configurable responses to link errors: data error to target abort; master abort to no error; configuration space master abort to no error; sync flood on data errors; sync flood on target aborts.
  - Added another MCA bank for a CPU watchdog timer.
  - MCA control mask registers control logging in addition to reporting of errors.
  - Link protocol checking.
  - Ability to convert machine check exceptions into vectored interrupts or SMIs.
- General Northbridge additions:
  - Support for an L3 cache, shared between cores, in product variations.
  - Probe filter to improve performance for revision D and later multi-processor systems.
  - BIOS-initiated system memory clear command.
  - MMIO-based access to configuration space and support for extended configuration space; this includes support for SMI traps to these accesses as well.
  - SMBus-based access to internal processor state called sideband interface (SBI) also referred to as APMML for revision D and later.
  - Mode whereby the IO request response order matches the IO request order.
  - VGA space decoding to MMIO-space mapping registers.



- Support for multiple, simultaneous GART/DEV table walks.
- Support for evenly distributed traffic in systems that connect multiple links between the same processors.
- More DEV protection domains and a larger DEV cache.
- Ability to support IOMMU chipsets (using IFCM) for revision D and later.
- Ability to force all IO requests to system memory (except display refresh) to probe the cache, in support of nested paging.
- Combined link status register for all links.
- Multi-node processor support for revision D and later.
- Power management:
  - Simple “fire and forget” operating system interface for P-state changes.
  - Separate core and Northbridge power and clock planes.
  - PVI and SVI regulator interfaces.
  - Support for up to 5 independent P-states for each core.
  - Support for 2 Northbridge P-states for revision DA-C.
  - Support for P-state limits controlled by sideband interface (SBI), thermal limits (HTC), or host software (STC); used to limit the P-state requested by the operating system in order to reduce power.
  - Core performance boost for revision E.
  - Support for core alternate voltage (C5) for revision C3.

### 1.5.3 Changes For Revision C

See the following references for information about changes for this revision.

- Changes that may result in BIOS modifications.
  - [2.7.1.3.1 \[Unused Links\]](#).
  - [Table 86: support for 8Gb DDR3 devices.](#)
  - [F2x\[1, 0\]78\[EarlyArbEn\].](#)
  - [F2x\[1, 0\]9C\\_x\[3:0\]0\[2:1\].](#)
  - [F2x\[1, 0\]9C\\_x\[3:0\]03.](#)
  - [F2x\[1, 0\]9C\\_x\[3:0\]0\[6:5\].](#)
  - [F2x\[1, 0\]9C\\_x\[3:0\]07.](#)
  - [F2x\[1, 0\]9C\\_x0D.](#)
  - [F2x\[1, 0\]9C\\_x\[4A:30\]](#)
  - [F2x\[1, 0\]A8\[DataTxFifoWrDly\].](#)
  - [F2x110\[DctSelIntLvAddr\].](#)
  - [Table 100.](#)
  - [F3x180\[SyncFloodOnDramTempErr\].](#)
  - [F2x1B0.](#)
  - [F3x1E4\[LvtOffset\].](#)
  - [F3x1E8\[SbiBankSel, SbiByteCnt\].](#)
  - [F4x1\[9C, 94, 8C, 84\]\\_x\[D5, C5\]\[VML, MapPreToMargin, TxLs23ClkGateEn\].](#)
  - [F4x1\[9C, 94, 8C, 84\]\\_x\[530A, 520A\]\[DllAnalogOkIgnore\].](#)
  - [F4x1C4.](#)
  - [MSRC001\\_102A\[RdMmExtCfgQwEn\].](#)
- Other changes.
  - [F3x1EC\[SbiRegDat3, SbiRegDat2, SbiRegDat1\].](#)
  - [F4x1\[9C, 94, 8C, 84\]\\_x\[D4, C4\]\[VdcDacFine\].](#)
  - [F4x1\[9C, 94, 8C, 84\]\\_x\[530A, 520A\]\[AnalogWaitTime\].](#)
  - [F4x1\[9C, 94, 8C, 84\]\\_x6\[9, 8\]84.](#)
  - [F4x1\[9C, 94, 8C, 84\]\\_x\[D1, C1\]\[LfxMax, LfcMin\].](#)
  - [F4x1\[9C, 94, 8C, 84\]\\_x\[D0, C0\]\[RttIndex, RonIndex\].](#)

- CPUID Fn[8000\_0000, 0000\_0000]\_EAX[LFuncExt].
- CPUID Fn8000\_0001\_ECX[SKINIT, WDT].
- CPUID Fn8000\_000A[NRIPS].
- CPUID Fn8000\_0019.
- CPUID Fn8000\_001B.
- MSRC001\_1033[IbsOpCntCtl, IbsOpCurCnt].
- MSRC001\_1037[IbsDcL2tlbHit1G].
- EventSelect 041h.
- EventSelect 045h[3].

### 1.5.3.1 Changes for Revision DA-C

See the following references for information about changes for this revision.

- Changes that may result in BIOS modifications.
  - 2.4.2.2 [NB P-states].
  - Table 38.
  - Table 51.
  - F0x16C[InLnSt].
  - F2x[1, 0]90[DisDllShutdownSR].
  - Table 100.
  - F2x[1,0]9C\_x0D0F0[F:0]30.
  - F2x[1,0]9C\_x0D0F0[F:0]13.
  - F2x[1,0]9C\_x0D0F2[F:0]30.
  - F2x[1,0]9C\_x0D0FE006.
  - F2x[1,0]9C\_x0D0FE007.
  - F3xD4[NbClkDiv].
  - F3x188[EnStpGntOnFlushMaskWakeup].
  - F4x1[9C, 94, 8C, 84]\_x[D5, C5][TxLs23ClkGateEn].
  - MSRC001\_0055[C1eOnCmpHalt].
  - MSRC001\_001F[GfxNbPstateDis].
- Other changes.
  - F2x[1, 0][5C:40].
  - F2x[1, 0]88[MemClkDis].
  - F2x[1, 0]9C\_x0C[ODTTri, ChipSelTri].
  - F2x[1,0]9C\_x0D0F0[F,8:0]0[8,4,0].

### 1.5.3.2 Changes for Revision C3

See the following references for information about changes between revision C models.

- Changes that may result in BIOS modifications.
  - 2.4.3.5 [C5].
  - F0x16C[FullT0Time, InLnSt].
  - F2x10C.
  - F3x[84:80].
  - F3xA0[SviHighFreqSel].
  - F3xDC[AltvidVSSlamTime, SlamTimeMode].
  - F3x1F0[NbPstate]
  - MSRC001\_0055[C1eOnCmpHalt, IntrPndMsg, IntrPndMsgDis].



### 1.5.4 Changes For Revision D

See the following references for information about changes for this revision.

- Changes that may result in BIOS modifications.
  - 2.4.2.9 [BIOS Northbridge COF and VID Configuration].
  - 2.4.2.9.1 [BIOS NB COF and VID Configuration for SVI and Single-Plane PVI Systems].
  - 2.7.4 [Link Bandwidth Requirements].
  - 2.7.7 [LDTSTOP Requirements].
  - F0x6C[Tr0Double].
  - F0x[F0, D0, B0, 90].
  - F0x[FC, DC, BC, 9C].
  - F0x160[CpuCnt].
  - F0x168[Cpu5En, Cpu4En].
  - F0x1A0[IntLnkRoute].
  - F0x1E0.
  - F2x[1, 0]88[MemClkDis].
  - F2x[1, 0]94[MemClkFreq].
  - F2x[1, 0]A4.
  - F2x[1, 0]A8[ProgOdtEn].
  - F2xAC.
  - F2x[1, 0]F0, F2x[1, 0]F4.
  - F2x11C[PrefThreeConf, DctWrLimit, MctPrefReqLimit].
  - F2x1B0[DcqBwThrotWm, PrefFiveConf, PrefFourConf, SpecPrefThreshold, SpecPrefMis, SpecPrefDis, EnSplitDctLimits].
  - F3x44[WDTDis].
  - F3x78.
  - F3xC4.
  - F3xE8[MultiNodeCpu, IntNodeNum, CmpCap].
  - F3x138 and F3x13C.
  - F3x180[ChgUcToCeEn, EccSymbolSize, McaLogErrAddrWdtErr].
  - F3x1A0[L3FreeListCBC, CpuCmdBufCnt].
  - F3x1D4.
  - F3x1FC.
  - F4x1[9C, 94, 8C, 84]\_x[D4, C4][DfeEn, DfeVoltage].
  - F4x1[9C, 94, 8C, 84]\_x144[ErrCntCtl].
  - F4x1[9C, 94, 8C, 84]\_x148.
  - F4x1[9C, 94, 8C, 84]\_x14C.
  - F4x1[9C, 94, 8C, 84]\_x4[9:0][8, 0]6.
  - F4x1[9C, 94, 8C, 84]\_x6[9:0][8, 0]0.
  - F4x[FC, DC, BC, 9C].
  - MSRC001\_102A[ProbeFilterSupEn, ThrottleNbInterface].
- Other changes.
  - F2x[1, 0]9C\_x00[ProcOdt].
  - F2x[1,0]9C\_x0D0F0[F,8:0]0[8,4,0].
  - F2x[1,0]9C\_x0D0F0[F:0]13.
  - F2x[1,0]9C\_x0D0F0[F:0]30.
  - F2x[1,0]9C\_x0D0F2[F:0]30.
  - F2xF8 and F2xFC.
  - F3x4C[ErrCpuVal, ErrCpu].
  - F3x50 and Table 96.
  - F3xA4[CurTmp].

- F3x1E8[SbiByteCnt, SbiBankSel].
- F3x1EC[SbiRegDat3, SbiRegDat2, SbiRegDat1].
- F4x1C8.
- CPUID Fn8000\_000A\_EDX[PauseFilter].
- MSRC001\_001F[InitApicIdCpuIdLo].
- MSRC001\_0071[CurCpuVid].
- EventSelect 4E0h.
- EventSelect 4E1h.
- EventSelect 4E2h.
- EventSelect 4EDh.

#### 1.5.4.1 Changes for Revision D1

See the following references for information about changes for this revision.

- Changes that may result in BIOS modifications.
  - 2.4.3.3.5 [BIOS Requirements to Initialize Message Triggered C1E].
  - F0x[18C:170][LS2En].
  - F2x[1, 0]90[DisDllShutdownSR].
  - F3x[84:80].
  - F3xA0[BpPinsTriEn].
  - F4x1[9C, 94, 8C, 84]\_x[D5, C5][TxLs23ClkGateEn].

##### 1.5.4.1.1 Multi-node Processors

See the following references for information specific to multi-node processors.

- 2.6.4.2 [HyperTransport™ Technology Routing].
- 2.6.4.2.2 [NodeId Enumeration Requirements for Dual-node Processors].
- 2.7.1.5 [Link Mapping for Dual-Node Processors in the G34 Package].
- 2.8 [DRAM Controllers (DCTs)].
- 2.8.2 [DDR Pad to Processor Pin Mapping].
- 2.9.2 [CPU Cores and Downcoring].
- F0x[E8, C8, A8, 88][Freq[3:0]].
- F0x1A0[IntLnkRoute].
- F3x64.
- F3x68.
- F3xA4[CurTmp].
- F3xC4.
- F3xE8[IntNodeNum, MultiNodeCpu].
- F3x1E4.
- F3x1E8.
- F3x1EC.
- F4x1[9C, 94, 8C, 84]\_x[D5, C5].
- MSRC001\_001F[InitApicIdCpuIdLo].
- MSRC001\_0071[CurCpuVid].
- MSRC001\_100C.

#### 1.5.5 Changes for Revision E

See the following references for information about changes for this revision.

- Changes that may result in BIOS modifications.
  - 2.4.2.1.1 [Core Performance Boost (CPB)].

- F2x[1, 0]90[DisDIIShutdownSR].
- F2x10C.
- Table 100.
- F3xA4[PerStepTimeDn, TmpMaxDiffUp, PerStepTimeUp].
- F3xA8[PopDownPstate, CacheFlushPopDownEn].
- F3xDC[CacheFlushOnHaltTmr].
- F3x188[EnStpGntOnFlushMaskWakeup].
- F4x15C.
- F4x16C.
- F4x1[9C, 94, 8C, 84]\_x[D5, C5][TxLs23ClkGateEn].
- MSRC001\_0055.
- MSRC001\_0073.
- Other changes:
  - 2.4.2.1.2 [Core P-state Naming and Numbering].
  - 2.4.2.1.3 [Core P-state Control].
  - F3x1F0[MaxSwPstateCpuCof].
  - CPUID Fn[8000\_0000, 0000\_0000]\_EAX[LFuncStd].
  - CPUID Fn0000\_0006.
  - CPUID Fn8000\_0007[CPB].
  - MSR0000\_00E7.
  - MSR0000\_00E8.
  - MSRC001\_0015[EffFreqCntMwait, CpbDis].
  -

### 1.5.6 Supported Feature Variations

The following table specifies the first revision of the processor that is expected to be productized for each feature (blank entries indicate that the feature is not supported) and the first revision of the processor a feature is expected to be removed.

**Table 2: Supported feature variations by revision**

Feature	First Revision Supported	First Revision Removed
6 cores in a single node	D	
Unbuffered DDR2 DIMM interface up to 800 MT/s	B	
Unbuffered DDR2 DIMM interface up to 1067 MT/s	B	
Unbuffered DDR3 DIMM interface up to 1333 MT/s	C	
Unbuffered DDR2/DDR3 SO-DIMMs	C	D
Registered DDR2 DIMM interface up to 800 MT/s	B	
Registered DDR3 DIMM interface up to 1333 MT/s	D	
DRAM x4 ECC symbol size	B	
DRAM x8 ECC symbol size	D	
DRAM thermal management and power capping	D	
Low voltage DDR3 1.35V support up to 1067 MT/s	C3	
Low voltage DDR3 1.35V support up to 1333 MT/s	D1	
Gen3 link and retry DC-coupled mode (coherent links)	C	
Gen3 link and retry DC-coupled mode (non-coherent links)	B	
Link decision feedback equalization	D	
Link unganging support	D	
Narrow (2-bit and 4-bit) link		B
CRC insertion on Gen3 coherent links	C	
Cache flush on Halt	C	
Probe filter	D	
SMBus-based sideband thermal sensor interface (SB-TSI)	B	
SMBus-based sideband remote management interface (SB-RMI)	D	
Core performance boost	E	
Northbridge P-states	DA-C and C3	D
Single-plane compatible	B	D
Dual-plane compatible	B	
Triple-plane compatible		
SMI initiated C1E	B	
Hardware initiated C1E (dual-core devices only)	DA-C	C3
Hardware initiated C1E (single and multi-core devices)	C3, E	D
C3 (single core devices only)	DA-C	C3
C5/Altvid	C3	D
1. AMD recommends using PROCHOT_L for thermal throttling and not implementing stop clock based throttling.		

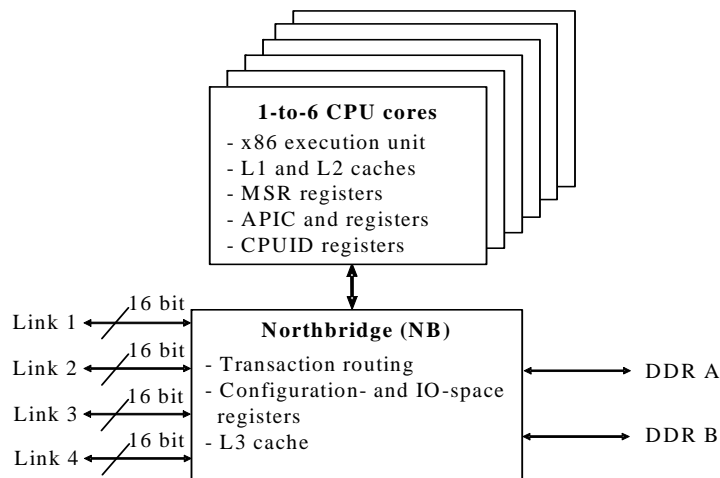
**Table 2: Supported feature variations by revision**

Feature	First Revision Supported	First Revision Removed
Message triggered C1E	D1	
Per-core IO-based C-states	E	
Thermal clock throttling (SMC controlled)	B <sup>1</sup>	
Isochronous flow control mode	C	
Display refresh traffic	B	D
DEV capability	B	D
IOMMU compatible	D	
3.4 MHz high speed SVI mode	C3	D
1. AMD recommends using PROCHOT_L for thermal throttling and not implementing stop clock based throttling.		

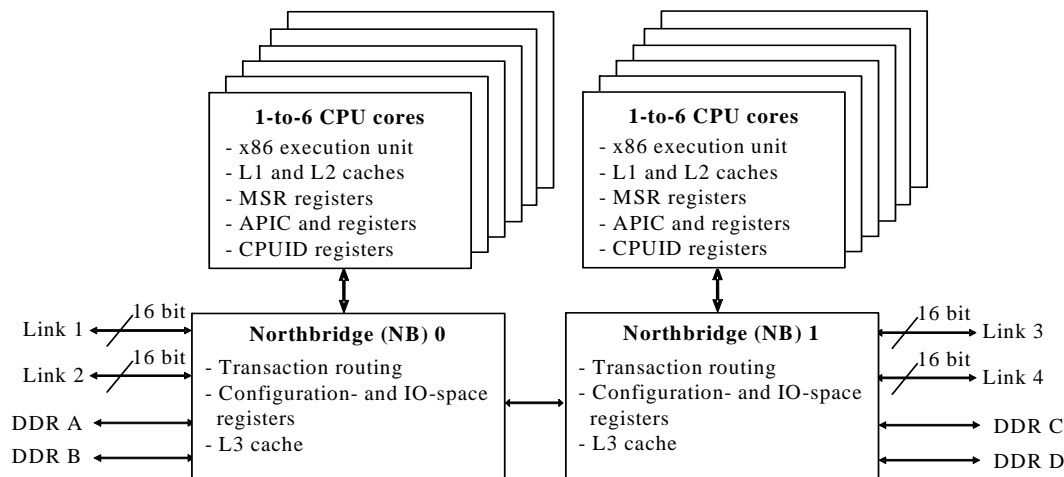
## 2 Functional Description

### 2.1 Processor Overview

The *processor* is a package that contains one or more nodes. A *node*, is an integrated circuit device that includes (1) one to six cores, (2) up to four links for general-purpose communication to other devices, (3) one or two 64-bit DDR DRAM interfaces for communication to system memory, and (4) one communication packet routing block referred to as the *Northbridge* (NB).



**Figure 1: A single-node processor.**



**Figure 2: A dual-node processor.**

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

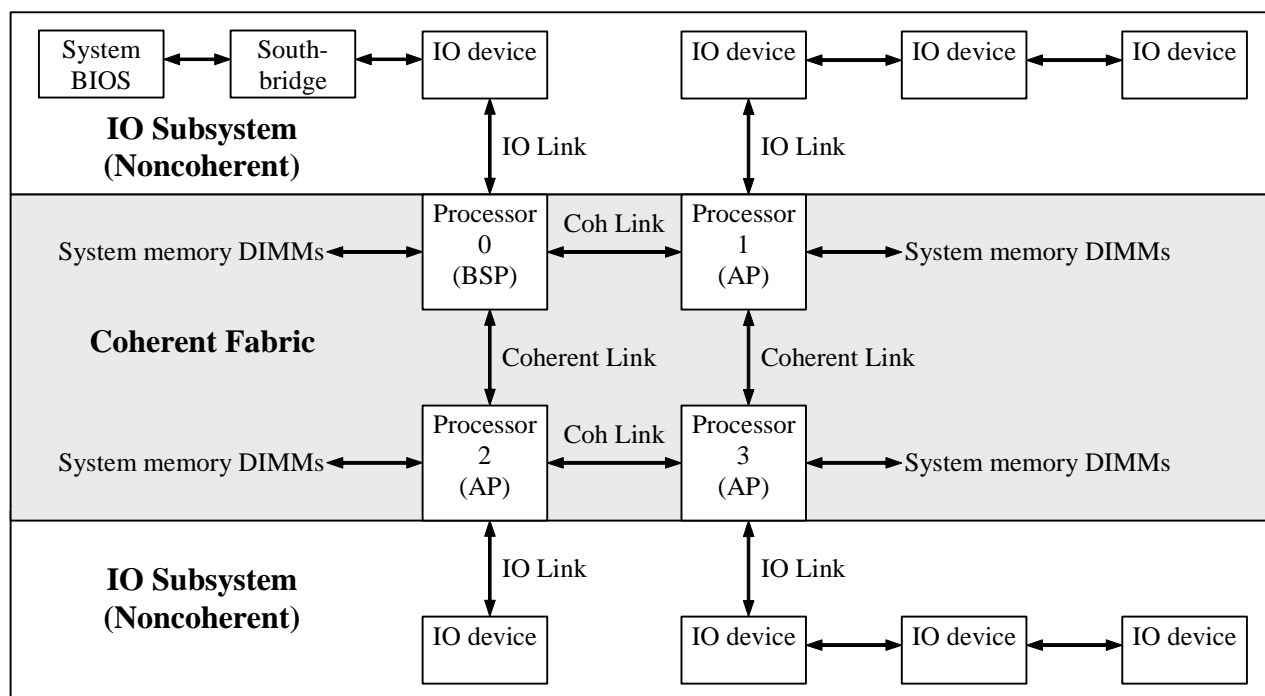
Each link can be configured to operate under the rules of one of the following interface specifications: (1) AMD proprietary, coherent inter-processor link; or (2) non-coherent HyperTransport™ IO link. When a link is configured for non-coherent IO traffic, it is referred to as an *IO link*.

Each DRAM interface supports a 64-bit DDR2 or DDR3 registered or unbuffered DIMM channel.

The NB routes transactions between the cores, the links, and the DRAM interfaces. It includes the configuration register space for the device. It may include an L3 cache as well.

## 2.2 System Overview

The following diagram illustrates the expected system architecture. Smaller systems may not include multiple processors or multiple IO links. Larger systems may include many more processors. Each processor in the coherent fabric communicates with other processors through the coherent link protocol. Processors communicate with the IO subsystem through IO links.



**Figure 3: System diagram.**

### 2.2.1 Mixed Stepping

See the *Revision Guide for AMD Family 10h Processors* for supported configurations of mixed steppings of Family 10h processors in the same system. If an unsupported configuration is detected, BIOS should configure the BSP as a single processor system and signal an error as described in 2.6.4.2.3 [BIOS Requirements for Systems with Mixed Processor Families].

## 2.3 Processor Initialization

This section describes the initialization sequence after a cold reset.

The processor that is connected to the IO hub is the BSP. Core 0 of the BSP begins executing code from the reset vector. Core 0 on all other nodes do not fetch code until their [The Link Initialization Control Register] F0x6C[ReqDis] bit is cleared. The remaining cores do not fetch code until their enable bits are set ([The Link Transaction Control Register] F0x68 [Cpu1En] for core 1 and [The Extended Link Transaction Control Register] F0x168[Cpu2En,Cpu3En] for cores 2 and 3).

### 2.3.1 BSP initialization

The BSP must perform the following tasks as part of POST.

- Store BIST information from the EAX register into an unused processor register.
- If supported, determine the type of this reset. One method is to use [The Link Initialization Control Register] F0x6C[InitDet] bit. If this boot sequence was caused by an INIT then BIOS vectors away from the cold/warm reset initialization path.
- Determine type of startup using the [The Link Initialization Control Register] F0x6C [ColdRstDet] bit. If this is a cold reset then BIOS must clear the [MCi\_STATUS] MSRs (see Table 72). If this is a warm reset then BIOS may check for valid MCA errors and if present save the status for later use (see 2.12.1.4 [Handling Machine Check Exceptions]).
- Enable the cache, program the MTRRs for Cache-as-RAM and initialize the Cache-as-RAM, as described in 2.3.3 [Using L2 Cache as General Storage During Boot].
- Setup of APIC (2.9.5.1 [ApicId Enumeration Requirements]).
- Perform coherent link enumeration (routing table and Node ID), as described in 2.6.4.2 [HyperTransport™ Technology Routing].
- Configure all IO-link devices.
  - Set configuration-base and -limit ([The Configuration Map Registers] F1x[EC:E0] [BusNumBase], [BusNumLimit]) and assign BUID.
  - Device enumeration for all IO-link devices (see link specification).
- If required, reallocate data and flow control buffers of the links (see [The Link Base Channel Buffer Count Registers] F0x[F0, D0, B0, 90] and [The Link Isochronous Channel Buffer Count Registers] F0x[F4, D4, B4, 94]) and issue system warm reset.
- Configure links speed and link width (see link specification).
- Configure processor power management (see 2.4 [Power Management]).
- If supported, allow other cores to begin fetching instructions by clearing [The Link Initialization Control Register] F0x6C[ReqDis] in the PCI configuration space of all other nodes and setting [The Link Transaction Control Register] F0x68[Cpu1En] and [The Extended Link Transaction Control Register] F0x168[Cpu2En,Cpu3En] in the PCI configuration space of all nodes.

### 2.3.2 AP initialization

All other processor cores other than core 0 of node 0 begin executing code from the reset vector. They must perform the following tasks as part of POST.

- Store BIST information from the eax register into an unused processor register.
- If supported, determine the type of startup from either the keyboard controller or the [The Link Initialization Control Register] F0x6C[InitDet] bit. If this boot sequence was caused by an INIT then BIOS vectors away from the cold/warm reset initialization path.
- Determine the history of this reset using the [The Link Initialization Control Register] F0x6C [ColdRstDet] bit. If this is a cold reset then BIOS must clear the [MCi\_STATUS] MSRs (see Table 72). If this is a warm reset then BIOS may check for valid MCA errors and if present save the status for use later (see 2.12.1.4 [Handling Machine Check Exceptions]).
- Setup of local APIC (2.9.5.1 [ApicId Enumeration Requirements]).
- Configure processor power management (see 2.4 [Power Management]).

### 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 of each core as gen-



eral storage. BIOS manages the mapping of the L2 storage such that cacheable accesses do not cause L2 victims.

The L2 cache as storage is described as follows:

- Each core has its own L2 cache.
- BIOS manages the mapping of the L2 storage such that cacheable accesses do not cause L2 victims.
- The L2 size, L2 associativity, and L2 line size is determined by reading [CPUID Fn8000\\_0006\\_ECX](#)[L2Size, L2Assoc, L2LineSize]. L2WayNum is defined to be the number of ways indicated by the L2Assoc code.
  - The L2 cache is viewed as (L2Size/L2LineSize) cache lines of storage, organized as L2WayNum ways, each way being (L2Size/L2WayNum) in size.
    - E.g. L2Assoc=8 so L2WayNum=16 (there are 16 ways). If L2Size=512KB then there are 16 blocks of cache, each 512KB/16 in size, or 32KB each.
  - For each of the following values of L2Size, the following values are defined:
    - L2Size=512KB: L2Tag=PhysAddr[39:15], L2WayIndex=PhysAddr[14:6].
    - L2Size=1MB: L2Tag=PhysAddr[39:16], L2WayIndex=PhysAddr[15:6].
  - 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.
- BIOS can rely on a minimum L2Size of 512 KB. See [CPUID Fn8000\\_0006](#)[L2Size].

The following memory types are supported as follows:

- WP-IO: BIOS ROM may be assigned the write-protect IO memory type and may be accessed read-only as data and fetched as instructions.
  - BIOS initializes a location in the L2 cache, mapped as write-protect IO, with 1 load of any size or an instruction fetch to any location within the L2LineSize cache line.
- WB-DRAM: General storage may be assigned the write-back DRAM memory type and may be accessed as read-write data, but not accessed by instruction fetch.
  - BIOS initializes an L2LineSize sized and aligned location in the L2 cache, mapped as write-back DRAM, with 1 read to at least 1 byte of the L2LineSize sized and aligned WB-DRAM address. BIOS may store to a line only after it has been allocated by a load.
  - Fills, sent to the disabled memory controller, return undefined data.
- All of memory space that is not accessed as WP-IO or WB-DRAM space must be marked as UC memory type.
- In order to prevent victimizing L2 data, no more than L2WayNum cache lines accessed as WP-IO or WB-DRAM may have the same L2WayIndex.
  - Software does not need to know which ways the L2WayNum lines are allocated to for any given value of L2WayIndex, only that invalid ways will be selected for allocation before valid ways will be selected for allocation.
  - Software can deallocate a line in the L2 by using CLFLUSH, and thus allow for a cache line to be filled with a different location. The following rules must be followed if CLFLUSH is used:
    - Place MFENCE instruction before CLFLUSH.
    - Don't jump to a CLFLUSH.

Performance monitor event [EventSelect 07Fh](#)[1], titled “L2 Writebacks to system“, can be used to indicate whether L2 dirty data was victimized and sent to the disabled memory controller.

The following requirements must be satisfied prior to using the cache as general storage:

- Paging must be disabled.

- `MSRC001_0015[INVDWBINVD]=0`.
- `MSRC001_1021[DIS_IND]=1`.
- `MSRC001_1021[DIS_SPEC_TLB_RLD]=1`. Disable speculative ITLB reloads.
- `MSRC001_1022[DIS_SPEC_TLB_RLD]=1`. Disable speculative DTLB reloads.
- `MSRC001_1022[DIS_CLR_WBTOL2_SMC_HIT]=1`.
- `MSRC001_1022[DIS_HW_PF]=1`.
- `MSRC001_102A[IcDisSpecTlbWr]=1`.
- `MSRC001_102A[CILinesToNbDis]=1`.
- INVD, and WBINVD must not be used during cache as general storage but may be used when tearing down cache-as-ram for all cores on a node.
- The BIOS must not use 3DNow!™, SSE, or MMX™ instructions, with the exception of the following list: MOVD, MOVQ, MOVDQA, MOVQ2DQ, MOVDQ2Q.
- The BIOS must not enable exceptions, page-faults, and other interrupts.
- BIOS must not use software prefetches.

When the BIOS is done using the cache as general storage the following steps are followed:

1. An INVD instruction should be executed on each core that used cache as general storage.
2. If DRAM is initialized and there is data in the cache that needs to get moved to main memory, CLFLUSH or WBINVD may be used instead of INVD, but software must ensure that needed data in main memory is not overwritten.
3. Restore the following configuration state:
  - `MSRC001_0015[INVD_WBINVD]`.
  - `MSRC001_1021[DIS_IND]=0`.
  - `MSRC001_1021[DIS_SPEC_TLB_RLD]=0`.
  - `MSRC001_1022[DIS_SPEC_TLB_RLD]=0`.
  - `MSRC001_1022[DIS_CLR_WBTOL2_SMC_HIT]=0`.
  - `MSRC001_1022[DIS_HW_PF]=0`.
  - `MSRC001_102A[IcDisSpecTlbWr]=0`.

When BIOS is done executing from WP-IO the following steps are followed:

1. `MSRC001_102A[CILinesToNbDis]=0`.

### 2.3.4 Multiprocessing Capability Detection

The multiprocessing capability of the processor is determined by `F3xE8[MpCap]`.

During POST, the BIOS checks the multiprocessing capability of all processors, and configures the system accordingly.

Multiprocessing capability detection is not required in a single processor system.

All processors must be dual-processor (DP) capable or multiprocessor (MP) capable in a DP system. If any processor is not at least DP capable, the BIOS must configure the BSP as a uni-processor (UP), and must not initialize the AP.

All processors must be MP capable in an MP system. If any processor is not MP capable, the BIOS must configure the BSP as a UP processor, and must not initialize APs.

If all processors do not have adequate multiprocessing capability for a DP or an MP system, the BIOS must display the following message:

\*\*\*\*\* Warning: non-MP Processor \*\*\*\*\*

The processor(s) installed in your system are not multiprocessing capable. Now your system will halt.

If all processors have adequate multiprocessing capability for a DP or an MP system, but have different model numbers or operate at different frequencies, see [2.4.2.12 \[Mixed-Frequency and Power P-State Configuration\]](#).

### 2.3.5 BIOS Requirements For 64-Bit Operation

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

### 2.3.6 SLIT and SRAT

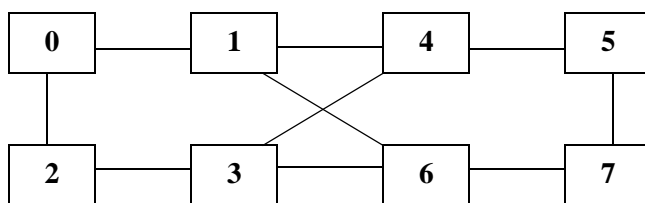
The System Locality Distance Information Table (SLIT) and Static Resource Affinity Table (SRAT) are described in the *Advanced Configuration and Power Interface Specification*.

#### 2.3.6.1 SLIT

The SLIT table is programmed with the following requirements:

- The local node in the SLIT table is 10.
- For fully connected system topologies, the remaining table entries are 16. A fully connected system topology is one where the number of hops between any two nodes in the system is one.
- For system topologies that are not fully connected:
  - If probe filter is enabled (F3x1D4[PfMode]=10b), remaining table entries are programmed with the formula  $10 + \text{num\_hops} * 6$ .
  - If probe filter is disabled (F3x1D4[PfMode]=00b), remaining table entries are programmed such that the maximum hop entries have a value of 13 and all other entries have 10.

Consider the 8 node system topology in [Figure 4](#). The maximum number of hops between any two nodes is 3. [Table 3](#) and [Table 4](#) show the SLIT table entries without probe filter and with probe filter. Note the values are normalized latencies and not absolute latencies. The raw latency of a system without the probe filter enabled is higher than one with the probe filter enabled.



**Figure 4: Example 8 node system in twisted ladder topology**

**Table 3: SLIT table example without probe filter**

Node	0	1	2	3	4	5	6	7
0	10	10	10	10	10	13	10	13
1	10	10	10	10	10	10	10	10
2	10	10	10	10	10	13	10	13

3	10	10	10	10	10	10	10	10
4	10	10	10	10	10	10	10	10
5	13	10	13	10	10	10	10	10
6	10	10	10	10	10	10	10	10
7	13	10	13	10	10	10	10	10

**Table 4: SLIT table example with probe filter**

Node	0	1	2	3	4	5	6	7
0	10	16	16	22	22	28	22	28
1	16	10	22	22	16	22	16	22
2	16	22	10	16	22	28	22	28
3	22	22	16	10	16	22	16	22
4	22	16	22	16	10	16	22	22
5	28	22	28	22	16	10	22	16
6	22	16	22	16	22	22	10	16
7	28	22	28	22	22	16	16	10

### 2.3.6.2 SRAT

A unique proximity domain is assigned for each node in the system. A processor local APIC affinity structure is created such that all cores in a node are assigned the same proximity domain as the node. The base address and length of the memory attached to the node is programmed into a memory affinity structure. See the *Advanced Configuration and Power Interface Specification* for additional information.

## 2.4 Power Management

The processor supports operational performance states, called P-states, ACPI power-savings states, and HTC. Processor power consumption may be altered in any of these states through control over clocking and voltage. Operational states are defined as states in which the processor is executing instructions, running software. During ACPI power-saving states, the processor does not execute instructions. [Table 5](#) provides a summary of each power management state and indicates whether it is supported. Refer to [F3x\[84:80\] \[ACPI Power State Control Registers\]](#) for processor configuration settings for power management.

**Table 5: Power management support**

ACPI/Power Management State	Supported <sup>1</sup>	Description
G0/S0/C0: Working	Yes	
G0/S0/C0: NB P-state transitions	Revision Specific	<a href="#">2.4.2.2 [NB P-states]</a>
G0/S0/C0: Core P-state transitions under OS control	Yes	<a href="#">2.4.2.1 [Core P-states]</a>
<ol style="list-style-type: none"> <li>Entries in the 'Supported' column indicate the following: <ul style="list-style-type: none"> <li>'Yes' indicates the described ACPI state is supported in all packages.</li> <li>'No' indicates the described ACPI state is not supported in any package.</li> <li>'Revision Specific' indicates support for the described ACPI state varies by processor revision and is covered in <a href="#">Section 1.5.6 [Supported Feature Variations]</a>.</li> </ul> </li> <li>AMD recommends using PROCHOT_L for thermal throttling and not implementing stop clock based throttling.</li> <li>S1 is not supported for processors that support NB P-states (<a href="#">F3x1F0[NbPstate] != 000b</a>).</li> </ol>		

**Table 5: Power management support**

ACPI/Power Management State	Supported <sup>1</sup>	Description
G0/S0/C0: Hardware thermal control (HTC)	Yes	2.10.3.1 [PROCHOT_L and Hardware Thermal Control (HTC)]
G0/S0/C0: Thermal clock throttling (SMC controlled)	Yes <sup>2</sup>	
G0/S0/C1: Halt	Yes	
G0/S0/C2: Stop-grant Caches snoopable	No	
G0/S0/C3: Stop-grant Caches not snoopable (single-core devices only)	Revision Specific	
G0/S0/C1E: Stop-grant Caches not snoopable using MSRC001_0055[C1eOnCmpHalt] (dual-core devices only)	Revision Specific	2.4.3.3.3 [Hardware Initiated C1E]
G0/S0/C1E: Stop-grant Caches not snoopable using MSRC001_0055[C1eOnCmpHalt] (single and multi-core devices)	Revision Specific	2.4.3.3.3 [Hardware Initiated C1E]
G0/S0/C1E: Stop-grant Caches not snoopable using MSRC001_0055[SmiOnCmpHalt] (single and multi-core devices)	Yes	2.4.3.3.1 [SMI Initiated C1E]
G0/S0/C1E: Stop-grant Caches not snoopable using F3xD4[MTC1eEn] (single and multi-link devices)	Revision Specific	2.4.3.3.4 [Message Triggered C1E]
G0/S0/C1 Cache flushing during halt	Revision Specific	2.4.3.4 [Cache Flush On Halt]
G0/S0/Per-core IO-based C-states	Revision Specific	2.4.3.2 [C-state Request Interface]
G1/S1: Stand By (Powered On Suspend)	Yes <sup>3</sup>	
G1/S3: Stand By (Suspend to RAM)	Yes	2.4.4 [ACPI Suspend to RAM State (S3)]
G1/S4, S5: Hibernate (Suspend to Disk), Shut Down (Soft Off)	Yes	
G3 Mechanical Off	Yes	

- Entries in the ‘Supported’ column indicate the following:
  - ‘Yes’ indicates the described ACPI state is supported in all packages.
  - ‘No’ indicates the described ACPI state is not supported in any package.
  - ‘Revision Specific’ indicates support for the described ACPI state varies by processor revision and is covered in Section 1.5.6 [Supported Feature Variations].
- AMD recommends using PROCHOT\_L for thermal throttling and not implementing stop clock based throttling.
- S1 is not supported for processors that support NB P-states (F3x1F0[NbPstate] != 000b).

### 2.4.1 Processor Power Planes And Voltage Control

Refer to the *AMD Family 10h Processor Electrical Data Sheet* for power plane definitions.

The voltage level of VDD and VDDNB may be altered in various states to control power consumption. All the other supplies are fixed. Refer to the EDS for power plane sequencing requirements.

The processor includes two interfaces, intended to control external voltage regulators, called the parallel VID (voltage level identifier) interface (PVI) and the serial VID interface (SVI). The PVI is a simple 6-bit VID code provided on 6 pins. The SVI encodes voltage regulator control commands, including the VID code, using SMBus protocol over two pins, SVD and SVC, to generate write commands to external voltage regulators. The processor is the master and the voltage regulator(s) are the slave(s). Both pins are outputs of the master; SVD is driven by the slave as well. SVC is a clock that strobes the data pin, SVD, on the rising edge. The frequency of the SVC is controlled by F3xA0[SviHighFreqSel]. Refer to the AMD Design Guide for Voltage Regulator Controllers Accepting Serial VID Codes for details on SVI protocol. See 1.2 [Reference Documents].

The processor supports:

- Single-plane platforms in product variations. See Table 2. All the VDD and VDDNB power planes are connected together on the systemboard and controlled as a single power plane through the PVI (F3xA0[Pvi-

Mode]=1) interface.

- Dual-plane platforms in which the VDD and VDDNB planes are isolated on the systemboard and controlled as separate voltages through the SVI or PVI interface.

### 2.4.1.1 VID Pins And Interface Selection

The VID interfaces use pins VID[5:0]. While PWROK is deasserted, the processor tristates VID[1] so that it may be used to select the VID interface; VID[1] is expected to be strapped high or low through a resistor on the systemboard. When PWROK asserts, the processor samples VID[1] and captures the state in [The Power Control Miscellaneous Register] F3xA0[PviMode]. Processors that do not have VID[5:0] pins do not support PVI and F3xA0[PviMode] is always 0.

VID[5:0] are controlled as follows:

- If PWROK = 0, VID[1] is an input to the processor.
  - VID[5:2, 0] are in push-pull mode (outputs are driving high or low).
  - If VID[1] = 1, VID[5:2, 0] = the PVI boot VID value.
  - If VID[1] = 0:
    - VID[5:4, 0] = output a valid, undefined state.
    - VID[3:2] = the SVI boot VID value (see 2.4.1.6.3 [Serial VID (SVI) Encodings]).
- If PWROK = 1:
  - If F3xA0[PviMode] = 1: VID[5:0] are all driven and controlled as needed by the boot process.
  - If F3xA0[PviMode] = 0:
    - VID[5, 4, 1] are driven low.
    - VID[3] becomes the SVC pin of the SVI. VID[2] becomes the SVD pin of the SVI. Transition from push-pull mode to open-drain mode occurs some time between the assertion of PWROK and the first SVI command.

### 2.4.1.2 Internal VID Registers

The registers within the processor that contain VID fields all use 7-bit VID encodings (see Table 8), regardless of whether the processor is in SVI mode or PVI mode. Processor hardware translates to the 6-bit VID encoding if [The Power Control Miscellaneous Register] F3xA0[PviMode]=1. The translation from the 7-bit VID code, SviVid, to the 6-bit VID code, PviVid, is as follows:

```
If          SviVid == 7Fh through 5Eh, PviVid = 3Fh;                //0.375 volts
else if SviVid == 5Dh through 3Fh, PviVid = SviVid - 1Fh;
else if SviVid == 3Eh through 00h, PviVid[5:0] = SviVid[6:1];
```

- In a single-plane system the P-state VID is dictated by MSRC001\_00[68:64][NbVid] of the CPU-core in the highest-performance P-state.
- In a dual-plane system
  - The VID for VDDNB is dictated by MSRC001\_00[68:64][NbVid] of the CPU-core in the highest-performance P-state.
  - The VID for VDD is dictated by MSRC001\_00[68:64][CpuVid] of the CPU-core in the highest-performance P-state.

### 2.4.1.3 MinVid and MaxVid Check

The allowed limits of MinVid and MaxVid are provided in [The COFVID Status Register] MSRC001\_0071. Prior to generating VID-change commands to either the PVI or SVI, the processor filters the InputVid value to the OutputVid as follows:

- 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.

#### 2.4.1.4 PSI\_L

The processor supports indication of whether the processor is in a low-voltage state or not, which may be used by the regulator to place itself into a more power efficient mode. This is supported by the PSI\_L bit in the data field of the SVI command when F3xA0[PviMode]=0. It is enabled through F3xA0[PsiVidEn]. PSI\_L is asserted if the processor selects a VID code that is greater than or equal to (voltage that is less than or equal to) the VID code specified in F3xA0[PsiVid].

The voltage regulator must be able to supply the current required for the processor to operate at the VID code specified in F3xA0[PsiVid]. MSRC001\_00[68:64][IddValue, IddDiv] specify the maximum core current for each P-state. The following algorithm describes how to program PSI\_L on the core voltage planes.

```

After cold reset {
  If (F3xA0[PviMode]==0 and PSI is supported on the platform) {
    PSI_vrm_current = current supported by the VDD voltage regulator when PSI is enabled
    PSI_inrush_current = inrush current on the VDD plane during a voltage transition
    If (Revision == DA-C2 and CPUID Fn8000_0001_EBX[PkgType] == 0010b) {
      Previous_voltage = MSRC001_00[68:64][CpuVid] for the Pstate specified by
        MSRC001_0061[PstateMaxVal]-1
      Pstate_number_init = MSRC001_0061[PstateMaxVal]
    }
    Else {
      Previous_voltage = 7Fh
      Pstate_number_init = 0
    }
  }
  For (Pstate_number = Pstate_number_init; Pstate_number <= F3xDC[PstateMaxVal];
    Pstate_number++) {
    Pstate_current = ProcIddMax for the Pstate specified by Pstate_number. See
      2.4.2.11 [Processor-Systemboard Power Delivery Compatibility Check].
    If ((Pstate_number+1) > F3xDC[PstateMaxVal])
      Next_Pstate_current = 0;
    Else Next_Pstate_current = PSI_inrush_current + ProcIddMax for the Pstate specified
      by Pstate_number+1. See 2.4.2.11 [Processor-Systemboard Power Delivery Compatibility Check].
    Pstate_voltage = MSRC001_00[68:64][CpuVid] for the Pstate specified by Pstate_number
    If ((Pstate_current <= PSI_vrm_current) && ((Next_Pstate_current <= PSI_vrm_current)
      && (Pstate_voltage != Previous_voltage)) {
      Set F3xA0[PsiVid] = Pstate_voltage
      Set F3xA0[PsiVidEn] = 1
      Break
    }
    Previous_voltage = Pstate_voltage
  }
}

```

#### 2.4.1.5 Alternative Voltage (Altvid)

In order to save power, a lower alternative voltage (altvid) is applied in the C5 state. Altvid is only supported as part of the C5 state. See 2.4.3.5 [C5]. Altvid is only applied to the cores and is controlled by F3x[84:80][AltVidEn], F3xD8[AltVidStart] and F3xDC[AltvidVSSlamTime, SlamTimeMode, AltVid]. Feature support var-



ies by revision. See [Table 2](#).

Altvids are not supported in multi-node, multi-link, or PVI systems.

#### 2.4.1.6 VID Encodings

The following sections provide VID encoding to VDD translations. Section [2.4.1.6.1 \[Boot VID Encodings\]](#) defines the VID to VDD translation for both protocols prior to PWROK assertion. Sections [2.4.1.6.2 \[Parallel VID Interface \(PVI\) Encodings\]](#) and [2.4.1.6.3 \[Serial VID \(SVI\) Encodings\]](#) define the VID to VDD translation following PWROK assertion and protocol initialization (if required).

##### 2.4.1.6.1 Boot VID Encodings

Prior to PWROK assertion the VID pins drive the Boot VID value in the manner specified by section [2.4.1.1 \[VID Pins And Interface Selection\]](#). There are 4 possible VDD values that can be requested by the Boot VID. The following table shows the Boot VID to VDD translation for both SVI and PVI protocol.

In an SVI system each regulator is specified to drive a voltage corresponding to the Boot VID value after PWROK is asserted until an SVI command addressed to that regulator changes the requested voltage.

**Table 6: Boot VID codes**

PVI VID[5:0]	SVI VID[3:2]	VDD
01_0010b	00b	1.100
01_0110b	01b	1.000
01_1010b	10b	0.900
01_1110b	11b	0.800

##### 2.4.1.6.2 Parallel VID Interface (PVI) Encodings

The 6-bit VID code programmed into the regulator, PviVid[5:0], is expected to be encoded by the regulator as follows:

```
If PviVid >= 20h, voltage = 0.7625V - 0.0125V * (PviVid-20h);
else voltage = 1.550V - 0.025V * PviVid;
```

I.e, 12.5mV resolution from 0.3875V (3Eh) to 0.775V (1Fh) and 25mV resolution from 0.775V to 1.55V (00h). The following table provides the same information.

**Table 7: PVI VID codes**

VID[5:0]	VDD	VID[5:0]	VDD	VID[5:0]	VDD	VID[5:0]	VDD
00_0000b	1.550	01_0000b	1.150	10_0000b	0.7625	11_0000b	0.5625
00_0001b	1.525	01_0001b	1.125	10_0001b	0.7500	11_0001b	0.5500
00_0010b	1.500	01_0010b	1.100	10_0010b	0.7375	11_0010b	0.5375
00_0011b	1.475	01_0011b	1.075	10_0011b	0.7250	11_0011b	0.5250
00_0100b	1.450	01_0100b	1.050	10_0100b	0.7125	11_0100b	0.5125
00_0101b	1.425	01_0101b	1.025	10_0101b	0.7000	11_0101b	0.5000
00_0110b	1.400	01_0110b	1.000	10_0110b	0.6875	11_0110b	0.4875
00_0111b	1.375	01_0111b	0.975	10_0111b	0.6750	11_0111b	0.4750



**Table 7: PVI VID codes**

VID[5:0]	VDD	VID[5:0]	VDD	VID[5:0]	VDD	VID[5:0]	VDD
00_1000b	1.350	01_1000b	0.950	10_1000b	0.6625	11_1000b	0.4625
00_1001b	1.325	01_1001b	0.925	10_1001b	0.6500	11_1001b	0.4500
00_1010b	1.300	01_1010b	0.900	10_1010b	0.6375	11_1010b	0.4375
00_1011b	1.275	01_1011b	0.875	10_1011b	0.6250	11_1011b	0.4250
00_1100b	1.250	01_1100b	0.850	10_1100b	0.6125	11_1100b	0.4125
00_1101b	1.225	01_1101b	0.825	10_1101b	0.6000	11_1101b	0.4000
00_1110b	1.200	01_1110b	0.800	10_1110b	0.5875	11_1110b	0.3875
00_1111b	1.175	01_1111b	0.775	10_1111b	0.5750	11_1111b	0.3750

**2.4.1.6.3 Serial VID (SVI) Encodings**

The 7-bit VID code programmed into the regulator, SviVid[6:0], is expected to be encoded by the regulator as follows:

```

If SviVid[6:0] == 7Fh through 7Ch, voltage = 0V;
else voltage = 1.550V - 0.0125V * SviVid[6:0];

```

The following table provides the same information.

**Table 8: SVI and internal VID codes**

VID[6:0]	VDD	VID[6:0]	VDD	VID[6:0]	VDD	VID[6:0]	VDD
000_0000b	1.5500	010_0000b	1.1500	100_0000b	0.7500	110_0000b	0.3500
000_0001b	1.5375	010_0001b	1.1375	100_0001b	0.7375	110_0001b	0.3375
000_0010b	1.5250	010_0010b	1.1250	100_0010b	0.7250	110_0010b	0.3250
000_0011b	1.5125	010_0011b	1.1125	100_0011b	0.7125	110_0011b	0.3125
000_0100b	1.5000	010_0100b	1.1000	100_0100b	0.7000	110_0100b	0.3000
000_0101b	1.4875	010_0101b	1.0875	100_0101b	0.6875	110_0101b	0.2875
000_0110b	1.4750	010_0110b	1.0750	100_0110b	0.6750	110_0110b	0.2750
000_0111b	1.4625	010_0111b	1.0625	100_0111b	0.6625	110_0111b	0.2625
000_1000b	1.4500	010_1000b	1.0500	100_1000b	0.6500	110_1000b	0.2500
000_1001b	1.4375	010_1001b	1.0375	100_1001b	0.6375	110_1001b	0.2375
000_1010b	1.4250	010_1010b	1.0250	100_1010b	0.6250	110_1010b	0.2250
000_1011b	1.4125	010_1011b	1.0125	100_1011b	0.6125	110_1011b	0.2125
000_1100b	1.4000	010_1100b	1.0000	100_1100b	0.6000	110_1100b	0.2000
000_1101b	1.3875	010_1101b	0.9875	100_1101b	0.5875	110_1101b	0.1875
000_1110b	1.3750	010_1110b	0.9750	100_1110b	0.5750	110_1110b	0.1750
000_1111b	1.3625	010_1111b	0.9625	100_1111b	0.5625	110_1111b	0.1625
001_0000b	1.3500	011_0000b	0.9500	101_0000b	0.5500	111_0000b	0.1500
001_0001b	1.3375	011_0001b	0.9375	101_0001b	0.5375	111_0001b	0.1375
001_0010b	1.3250	011_0010b	0.9250	101_0010b	0.5250	111_0010b	0.1250
001_0011b	1.3125	011_0011b	0.9125	101_0011b	0.5125	111_0011b	0.1125
001_0100b	1.3000	011_0100b	0.9000	101_0100b	0.5000	111_0100b	0.1000

**Table 8: SVI and internal VID codes**

VID[6:0]	VDD	VID[6:0]	VDD	VID[6:0]	VDD	VID[6:0]	VDD
001_0101b	1.2875	011_0101b	0.8875	101_0101b	0.4875	111_0101b	0.0875
001_0110b	1.2750	011_0110b	0.8750	101_0110b	0.4750	111_0110b	0.0750
001_0111b	1.2625	011_0111b	0.8625	101_0111b	0.4625	111_0111b	0.0625
001_1000b	1.2500	011_1000b	0.8500	101_1000b	0.4500	111_1000b	0.0500
001_1001b	1.2375	011_1001b	0.8375	101_1001b	0.4375	111_1001b	0.0375
001_1010b	1.2250	011_1010b	0.8250	101_1010b	0.4250	111_1010b	0.0250
001_1011b	1.2125	011_1011b	0.8125	101_1011b	0.4125	111_1011b	0.0125
001_1100b	1.2000	011_1100b	0.8000	101_1100b	0.4000	111_1100b	0.0000
001_1101b	1.1875	011_1101b	0.7875	101_1101b	0.3875	111_1101b	0.0000
001_1110b	1.1750	011_1110b	0.7750	101_1110b	0.3750	111_1110b	0.0000
001_1111b	1.1625	011_1111b	0.7625	101_1111b	0.3625	111_1111b	0.0000

#### 2.4.1.7 BIOS Requirements for Power Plane Initialization

- In single-plane systems BIOS is required to place the lower VID code (higher voltage) specified in the [MSRC001\\_00\[68:64\]\[NbVid and CpuVid\]](#) fields into both of these fields. Repeat this operation for all enabled P-states.
- Configure [F3xA0\[SlamVidMode\]](#) and [F3xD8\[VSRampTime or VSSlamTime\]](#) based on the platform requirements.
- Configure [F3xD4\[PowerStepUp, PowerStepDown\]](#).
- Optionally configure [F3xA0\[PsiVidEn and PsiVid\]](#). Refer to section [2.4.1.4 \[PSI\\_L\]](#) for additional details.

#### 2.4.1.8 Hardware-Initiated Voltage Transitions

VDD and VDDNB voltage levels may be transitioned during state changes involving boot, reset, P-state, and stop-grant. Voltage transitions may be slammed or stepped as specified by [\[The Power Control Miscellaneous Register\] F3xA0\[SlamVidMode\]](#). [F3xA0\[SlamVidMode\]=1](#) means the processor sends a single VID code to the voltage regulator. The voltage stabilization time is specified by [F3xD8\[VSSlamTime\]](#) and, if altvid is enabled, [F3xDC\[AltvidVSSlamTime, SlamTimeMode\]](#). [F3xA0\[SlamVidMode\]=0](#) means the processor sends multiple VID codes one increment at a time until the destination VID is reached. The amount of time between each VID code is specified by [F3xD8\[VSRampTime\]](#). Voltages are transitioned according to [Table 9](#).

**Table 9: Voltage transition behavior**

First state	Second state	Voltage transition
Voltage off	PWROK assert	Voltages slammed to the factory-specified boot voltage.
PWROK assert	RESET_L deassert	Voltages slammed to the VID specified by <a href="#">MSRC001_0071[StartupPstate]</a> .
Any P-state	Any P-state	Voltage transitioned to the new P-state VID.
Any P-state	Alternate VID	Voltage transitioned to value programmed into <a href="#">F3xDC[AltVid]</a> .
Alternate VID	Any P-state	Voltage transitioned to the P-state prior to application of the alternate VID.

#### 2.4.1.9 Software-Initiated Voltage Transitions

The processor supports direct software VID control using [\[The COFVID Control Register\] MSRC001\\_0070](#). The setting for [F3xA0\[SlamVidMode\]](#) determines the sequence used for direct VID control. Hardware P-state

transitions using [The P-State Control Register] MSRC001\_0062 result in unpredictable behavior if software modifies the NbVid or CpuVid from the appropriate settings for the current P-state reported in [The P-State Status Register] MSRC001\_0063. If F3xA0[PviMode]=1b only changes to NbVid are driven on the PVI interface as defined in Section 2.4.1 [Processor Power Planes And Voltage Control].

#### 2.4.1.9.1 Software-Initiated NB Voltage Transitions

NewNbVid = the destination NB VID.

F3xA0[SlamVidMode]=1:

1. Write NewNbVid to all copies of MSRC001\_0070[NbVid].
2. Wait the specified F3xD8[VSSlamTime].

F3xA0[SlamVidMode]=0:

1. If NewNbVid > MSRC001\_0071[CurNbVid] write MSRC001\_0071[CurNbVid] + 1 to all copies of MSRC001\_0070[NbVid]; else write MSRC001\_0071[CurNbVid] - 1 to all copies of MSRC001\_0070[NbVid].
2. Wait the specified F3xD8[VSRampTime].
3. If MSRC001\_0071[CurNbVid] != NewNbVid goto step 1.

#### 2.4.1.9.2 Software-Initiated CPU Voltage Transitions

NewCpuVid = the destination CPU VID.

F3xA0[SlamVidMode]=1:

1. Write NewCpuVid to MSRC001\_0070[CpuVid].
2. Wait the specified F3xD8[VSSlamTime].

F3xA0[SlamVidMode]=0:

Software must use the sequence for F3xA0[SlamVidMode]=0 defined in section 2.4.1.9.1 [Software-Initiated NB Voltage Transitions] to control the single-plane through NbVid.

#### 2.4.1.10 SVI Protocol

The SVI protocol is specified in the AMD Voltage Regulator Specification, with the following exception:

- For all revisions except C3, only a 400kHz bus clock is supported. For revision C3, a 400 kHz or 3.4 MHz bus clock is supported. See F3xA0[SviHighFreqSel].

### 2.4.2 P-states

P-states are operational performance states (states in which the processor is executing instructions, running software) characterized by a unique frequency and voltage. The processor supports up to 5 P-states called P-states 0 through 4 or P0 through P4. P0 is the highest power, highest performance P-state; each ascending P-state number represents a lower-power, lower performance P-state than the prior P-state number. As P-state numbers increase, the operating frequency and voltage for a given P-state must be less than or equal to the frequency and voltage of the prior P-state. At least one enabled P-state (P0) is specified for all processors.

The processor supports dynamic P-state changes in independently-controllable frequency planes for each core and the NB; and independently-controllable voltage planes VDD and VDDNB. Refer to section 2.4.1 [Processor Power Planes And Voltage Control] for voltage plane definitions and section 1.5.6 [Supported Feature Variations] for package/socket-specific information on voltage plane compatibility.

The following terminology applies to P-state definitions:

- 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 may be used to reduce the operating frequency.
- COF: current operating frequency.
  - Refer to [2.4.2.1 \[Core P-states\]](#) for details on the reference clock frequency and allowed DIDs for core P-states. Refer to [MSRC001\\_00\[68:64\]\[CpuFid\]](#) for the CPU COF formula and details on allowed FIDs for core P-states.
  - Refer to [F3xD4\[NbFid\]](#) for the NB COF formula.
- MOF: maximum operating frequency. This is the maximum operating frequency that software can request. This is specified as the COF of P0 found in [MSRC001\\_0064 \(MSRC001\\_00\[68:64\]\)](#) and [F3xD4\[NbFid\]](#) (for the NB) after a cold reset. This uses software P-state numbering. See [2.4.2.1.2.1 \[Software P-state Numbering\]](#).
- VID: voltage ID. Specifies the voltage level for a given domain. Refer to [2.4.1.6 \[VID Encodings\]](#) for encodings.

Out of cold reset, the VID and FID of the NB and cores is specified by [\[The Power Control Miscellaneous Register\] F3xA0\[CofVidProg\]](#) and [\[The COFVID Status Register\] MSRC001\\_0071\[StartupPstate\]](#). [MSRC001\\_0071\[StartupPstate\]](#) always points to the minimum P-state supported by the processor.

The dynamic FID, DID, and VID values associated with P-state transitions for all frequency and voltage domains are specified by [\[The P-State \[4:0\] Registers\] MSRC001\\_00\[68:64\]](#). All FID and DID parameters must be programmed to equivalent values for all cores and NBs in the coherent fabric. Refer to the [MSRC001\\_00\[68:64\]](#) and [F3xD4\[NbFid\]](#) register definitions for further details on programming requirements. Processors with different default P-state definitions can be mixed in a multi-socket system and still satisfy the FID and DID programming requirements. Refer to section [2.4.2.12 \[Mixed-Frequency and Power P-State Configuration\]](#) for details on multi-socket, mixed-frequency and/or power initialization requirements.

### 2.4.2.1 Core P-states

Dynamic core P-state support is indicated by more than one enabled selection in [\[The P-State \[4:0\] Registers\] MSRC001\\_00\[68:64\]\[PstateEn\]](#). The FID, DID, and VID for each core P-state is specified in [\[The P-State \[4:0\] Registers\] MSRC001\\_00\[68:64\]](#). The COF for core P-states is a function of half the CLKIN frequency (nominally 100 MHz) and the DID may be 1, 2, 4, 8, and 16. Software controls the current core P-state request 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\[HwPstate\]=1b](#). P-state transitions using the hardware P-state control mechanism are not allowed until the P-state initialization requirements defined in section [2.4.2.6 \[BIOS Requirements for P-State Initialization and Transitions\]](#) are complete.

#### 2.4.2.1.1 Core Performance Boost (CPB)

Core Performance Boost (CPB) allows the processor to deterministically provide maximum performance while remaining within the specified power delivery and removal envelope. This allows P-states to be defined with higher frequencies and voltages than could be used without CPB. These P-states are referred to as boosted P-states.

- CPB is not supported on single-plane systems.
- Support for CPB is specified by [CPUID Fn8000\\_0007\[CPB\]](#).
- CPB is enabled if all of the following conditions are true:
  - [F3xA8\[CacheFlushPopDownEn\] = 1](#).
  - [F3x188\[EnStpGntOnFlushMaskWakeUp\] = 1](#).

- `F4x15C[BoostSrc]` = 11b.
- `F4x15C[NumBoostStates]` = 1.
- `F4x16C[CstateCnt]` != 0h.
- `MSRC001_0015[CpbDis]` = 0 for all cores.
- All P-states, both boosted and non-boosted, are specified in `MSRC001_00[68:64]`.
- A maximum of one boosted P-state is supported and specified by `F4x15C[NumBoostStates]`.
  - Support for the boosted P-state varies by product.
- The boosted P-state is always higher performance than non-boosted P-states.
- To ensure proper operation, the boosted P-state should be hidden from the operating system. BIOS should not provide ACPI `_PSS` entries for the boosted P-state. See 2.4.2.13.2 [`_PSS (Performance Supported States)`] for details.

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

Since support for core performance boost varies from product to product, the mapping between `MSRC001_00[68:64]` and the indices used to request P-state changes or status also varies. In order to clarify this, two different numbering schemes are used.

##### 2.4.2.1.2.1 Software P-state Numbering

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

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

For example, if `F4x15C[NumBoostStates]` contains the values shown below, then the P-states would be named as follows:

**Table 10: P-state name to MSR address**

<code>F4x15C[NumBoostStates]=1</code>		<code>F4x15C[NumBoostStates]=0</code> or <code>F4x15C[NumBoostStates]</code> does not exist	
P-state Name	Corresponding MSR Address	P-state Name	Corresponding MSR Address
Pb0	MSRC001_0064	P0	MSRC001_0064
P0	MSRC001_0065	P1	MSRC001_0065
P1	MSRC001_0066	P2	MSRC001_0066
P2	MSRC001_0067	P3	MSRC001_0067
P3	MSRC001_0068	P4	MSRC001_0068

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

##### 2.4.2.1.2.2 Hardware P-state Numbering

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

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

### 2.4.2.1.3 Core P-state Control

Core P-states are dynamically controlled by software and are exposed through ACPI objects (refer to section 2.4.2.13 [ACPI Processor P-State Objects]). Software requests a core P-state change by writing a 3 bit index corresponding to the desired core 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. When software requests the P0 state on a processor that supports core performance boost (i.e. writes 000b to `MSRC001_0062[PstateCmd]`), hardware dynamically places the core into the highest-performance P-state possible as determined by core performance boost. See 2.4.2.1.1 [Core Performance Boost (CPB)] for details. Table 11 shows how P-state numbers are mapped to P-state registers.

**Table 11: P-state name to P-state index**

F4x15C[NumBoostStates]=1			F4x15C[NumBoostStates]=0 or F4x15C[Num-BoostStates] does not exist		
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	P0	0	MSRC001_0064
P0	0	MSRC001_0065	P1	1	MSRC001_0065
P1	1	MSRC001_0066	P2	2	MSRC001_0066
P2	2	MSRC001_0067	P3	3	MSRC001_0067
P3	3	MSRC001_0068	P4	4	MSRC001_0068

Hardware sequences the frequency and voltage changes necessary to complete a P-state transition as specified by 2.4.2.5 [P-state Transition Behavior] with no additional software interaction required. Core P-states are changed without interaction with the external chipset. `MSRC001_0063[CurPstate]` reflects the current frequency component (COF) of each core as a 3 bit index corresponding to the current P-state number. For revision D1 and earlier processors, if core 1 `MSRC001_0063[CurPstate] = 010b`, then core 1 is at the P2 COF specified by `MSRC001_0066[CpuFid and CpuDid]`. For revision E processors, if core 1 `MSRC001_0063[CurPstate] = 010b` and `F4x15C[NumBoostStates]=1`, then core 1 is at the P2 COF specified by `MSRC001_0067[CpuFid and CpuDid]`. If a core is in the software P0 state (i.e. if `MSRC001_0063[CurPstate]=0`), the frequency of the core could be the frequency specified by software P0 or any boosted P-state. To determine the frequency of a core, see 2.4.5 [Effective Frequency].

Hardware controls the VID for each voltage domain according to the highest requirement of the frequency domain(s) on each plane. The number of frequency domains in a voltage domain is package/platform specific. Refer to section 1.5.6 [Supported Feature Variations] for package/platform specific voltage plane support. For example the VID for a 4-core single-plane system must be maintained at the highest level required for all 5 frequency domains (4 cores and NB). Refer to section 2.4.2.5 [P-state Transition Behavior] for details on hardware P-state voltage control. Section 2.4.1.7 [BIOS Requirements for Power Plane Initialization] specifies the processor initialization requirements for voltage plane control.

### 2.4.2.2 NB P-states

Up to two NB P-states may be supported, NB P-state 0 and NB P-state 1. At least one enabled NB P-state (NB P-state 0) is specified for all processors. NB P-state 1 is always half the frequency of NB P-state 0 and is specified by `MSRC001_00[68:64][NbDid]=1` for any enabled P-state. The COF for NB P-states is specified by `F3xD4[NbFid]`. The NB VIDs associated with NB P-states 0 and 1 are specified by `F4x1F4[NbVid0, NbVid1]`. NB P-states are supported in single-node, single-link systems only. Feature support varies by revision. See [Table 2](#). See `F3x1F0[NbPstate]`.

#### 2.4.2.2.1 NB P-state Control

The NB P-state is controlled by hardware and is not exposed through ACPI objects. NB P-states may be dynamically controlled by system software via `MSRC001_001F[GfxNbPstateDis]`.

The Northbridge is placed in NB P-state 0 if any of the following are true:

- `MSRC001_001F[GfxNbPstateDis]=1`
- At least one core is in a P-state which specifies `MSRC001_00[68:64][NbDid]=0`

The Northbridge is placed in NB P-state 1 if all of the following are true:

- `MSRC001_001F[GfxNbPstateDis]=0`
- All cores are in a P-state which specify `MSRC001_00[68:64][NbDid]=1`

Changes in either the core P-state or `MSRC001_001F[GfxNbPstateDis]` will cause the NB P-state to be re-evaluated. The current NB P-state is specified by `MSRC001_0071[CurNbDid]`.

#### 2.4.2.3 P-state Limits

P-states may be limited to lower-performance values under certain conditions, including HTC and STC logic. Registers that control this are [\[The Hardware Thermal Control \(HTC\) Register\] F3x64\[HtcPstateLimit\]](#), [\[The Software Thermal Control \(STC\) Register\] F3x68\[StcPstateLimit\]](#), and, for revision D and later revisions, [\[The SBI P-state Limit Register\] F3xC4](#). The current limit is provided in [\[The P-State Current Limit Register\] MSRC001\\_0061\[CurPstateLimit\]](#). Changes to the `MSRC001_0061[CurPstateLimit]` can be programmed to trigger interrupts through `F3x64[Ps1ApicLoEn and Ps1ApicHiEn]`. In addition, the maximum value P-state, regardless of the source, is limited as specified in `MSRC001_0061[PstateMaxVal]`.

#### 2.4.2.4 P-state Bandwidth Requirements

- The frequency relationship of (core COF / NB COF)  $\leq 2$  must be maintained for all supported P-state combinations. E.g., a core P0 COF of 2.4 GHz could not be combined with a NB P0 COF of 1.0 GHz; the NB P0 COF would have to be 1.2 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 1.2 GHz or less.
- All core P-states are required to be defined such that (NB COF/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:
  - CPU COF  $\geq 400$ Mhz.
- NB COF  $\geq 2 * \text{MEMCLK}$  frequency. E.g., for DDR1333 support, NB COF must be 1.4 GHz or higher. (MEMCLK would be 667 MHz). Note if NB P-states are enabled, an NB P-state 0 of 2.8 GHz or higher is required to provide an NB P-state 1 of 1.4 GHz.
- NB COF  $\geq 600$ MHz.
- See [2.7.4 \[Link Bandwidth Requirements\]](#) for NB COF and link bandwidth requirements.



### 2.4.2.5 P-state Transition Behavior

P-state changes normally include a COF change and a VID change. If the P-state number is increasing (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. VID changes may be slammed or ramped; see 2.4.1.8 [Hardware-Initiated Voltage Transitions].

P-state changes that include VID changes may take 100's of microseconds to complete. Once the processor has initiated a VID change for a domain, it completes it regardless of what commands are received while the P-state change takes place. 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.

There is one set of P-state control registers in each core. Each core may independently request to enter a different P-state. When lower-performance P-states are requested, the logic reduces the COF of the core; however, if that core shares its power plane with another core, the VID cannot change until the other core's P-state is reduced. In addition, the NB P-state follows the P-state specified by the highest-performance core P-state. For example, assume there are two cores, both initially in P0 (along with the NB), and the NB is on a separate power plane:

- If a first command is issued to place core 0 into P2, then:
  - If the cores are on separate supplies, then core 0's COF and VID are changed to P2.
  - If the cores are on the same supply, then core 0's COF is placed into P2, but the VID does not change.
  - The NB P-state does not change.
- If a second command is issued placing core 1 into P4, then:
  - If the cores are on separate supplies, then core 1's COF and VID are changed to P4.
  - If the cores are on the same supply, then core 1's COF is changed to P4 and then the VID is changed to P2 (the VID of the highest-performance core P-state on that power plane).
  - The NB P-state is changed to the P-state specified by P2 (the highest-performance core P-state).
- If a third command is issued placing core 1 back into P0, then:
  - The NB P-state is changed back to the P-state specified by P0.
  - If the cores are on separate supplies, then core 1's COF and VID are changed back to P0.
  - If the cores are on the same supply, then the VID is changed to P0 and then CPU1's COF is changed to P0.

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

- Once a P-state change starts, the P-state state machine (PSSM) continues through completion unless interrupted by a PWROK deassertion or RESET\_L assertion. 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:
  3. If there is no P-state transition activity, then the cores and NB remain in the current P-state.
    - If a RESET\_L assertion interrupts a P-state transition, then the COF remains in its current state at the time RESET\_L is asserted (either the value of the old or the new P-state) and the VID remains in its current state (perhaps at a VID between the old and the new P-states, if the VID was being stepped). BIOS is required to transition to valid COF and VID settings after a warm reset according to the sequence defined in section 2.4.2.15 [BIOS COF and VID Requirements After Warm Reset].
    - If F3xD4[NbFid] has changed, then the new value is applied to the NB PLL on the assertion of RESET\_L. It is assumed that BIOS adjusts the NB VID to the appropriate value prior to the warm reset. See 2.4.1.9 [Software-Initiated Voltage Transitions].
- If F3xA0[PviMode]=1, the P-state VID is dictated by MSRC001\_00[68:64][NbVid] of the CPU-core in the highest-performance P-state.



- The OS controls the P-state through [The P-State Control Register] MSRC001\_0062, independent of P-state limits described in [The Hardware Thermal Control (HTC) Register] F3x64[HtcPstateLimit], [The Software Thermal Control (STC) Register] F3x68[StcPstateLimit], and, for revision D and later revisions, [The SBI P-state Limit Register] F3xC4. 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 core 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 [The P-State Current Limit Register] MSRC001\_0061[PstateMaxVal], regardless of whether it is a limit or OS-requested, then the PstateMaxVal is used instead.

#### 2.4.2.6 BIOS Requirements for P-State Initialization and Transitions

P-state transitions can be used only if they are supported by the processor and by the system. BIOS requirements are:

1. Configure the F3x[84:80] [ACPI Power State Control Registers] according to the settings in Table 100.
2. Configure the Northbridge COF and VID for each processor appropriately based on the sequence described in 2.4.2.9 [BIOS Northbridge COF and VID Configuration].
3. Perform the algorithm described in 2.4.2.7 [BIOS Configuration for Dual-plane Only Support].
4. Perform the algorithm described in 2.4.2.8 [BIOS Configuration for Asymmetric Boost].
5. Program MSRC001\_00[68:64][NbDid, NbVid] as described in F3x1F0[NbPstate].
6. Complete the 2.4.1.7 [BIOS Requirements for Power Plane Initialization].
7. Complete the 2.4.2.15.1 [Core P-State Voltage Alignment After Warm Reset].
8. Transition all parts to the minimum performance P-state using the algorithm detailed in 2.4.2.15.3 [Core Minimum P-State Transition Sequence After Warm Reset].
9. Complete the 2.4.2.11 [Processor-Systemboard Power Delivery Compatibility Check].
10. Determine the valid set of P-states:
  - Based on the sequence described in 2.4.2.12 [Mixed-Frequency and Power P-State Configuration] for multi-processor systems.
  - Based on the enabled P-states indicated in [The P-State [4:0] Registers] MSRC001\_00[68:64][PstateEn] for single-processor systems.
11. BIOS may transition each core in the system to any enabled P-state using MSRC001\_0062.
12. If P-states are not supported, as indicated by only one enabled selection in [The P-State [4:0] Registers] MSRC001\_00[68:64][PstateEn], then BIOS must not generate ACPI-defined P-state objects described in section 2.4.2.13 [ACPI Processor P-State Objects]. Otherwise, the ACPI objects should be generated to enable P-state support.

The following must also be completed before P-state transitions are allowed:

- If MSRC001\_00[68:64][CpuFid] is different between any two enabled P-states, the PLL lock time must be specified by [The Power Control Miscellaneous Register] F3xA0[PIILockTime].
- Configure F3xD4[NbClkDivApplyAll, NbClkDiv, and ClkRampHystSel].

#### 2.4.2.7 BIOS Configuration for Dual-plane Only Support

Some processors are only supported by the dual-plane infrastructure. BIOS performs the following algorithm to determine whether the processor has been placed into an acceptable infrastructure and, if necessary, to enable P-states associated with that infrastructure:

```
// Determine whether algorithm applies to this processor
if (CPUID Fn8000_0001_EBX[PkgType] == 0001b && (revision C or E) {
```

```

// Determine whether processor is supported in this infrastructure
if ((F3x1FC[DualPlaneOnly] == 1) && (this is a dual-plane platform)) {
    // Fixup the P-state MSRs
    for (each core in the system) {
        if (CPUID Fn8000_0007[CPB]){
            Copy MSRC001_0065 as MinPstate;
            Copy MSRC001_0068 to MSRC001_0065;
            Copy MinPstate to MSRC001_0068;
        } else {
            Copy MSRC001_0068 to MSRC001_0064;
            Program MSRC001_0068 = 0;
        } // endif
        for (each MSR in MSRC001_00[68:64]) {
            if (value in MSRC001_00[68:64][IddValue] != 0) {
                Set PstateEn in current MSR to 1;
            } // endif
        } // endfor
    } // endfor
    Set F3xDC[PstateMaxVal] = lowest-performance enabled P-state;
    Set F3x64[HtcPstateLimit] = lowest-performance enabled P-state;
} // endif
} // endif

```

#### 2.4.2.8 BIOS Configuration for Asymmetric Boost

Asymmetric boost allows certain cores to boost at a higher frequency. BIOS performs the following algorithm to assign these frequencies:

```

// Determine whether the processor support boost
if (CPUID CPUID Fn8000_0007[CPB]==1)&& CPUID Fn8000_0008[NC]==5){
    Core0 MSRC001_0064[CpuFid] += F3x10C[AsymmetricBoostCore0]
    Core1 MSRC001_0064[CpuFid] += F3x10C[AsymmetricBoostCore1]
    Core2 MSRC001_0064[CpuFid] += F3x10C[AsymmetricBoostCore2]
    Core3 MSRC001_0064[CpuFid] += F3x10C[AsymmetricBoostCore3]
    Core4 MSRC001_0064[CpuFid] += F3x10C[AsymmetricBoostCore4]
    Core5 MSRC001_0064[CpuFid] += F3x10C[AsymmetricBoostCore5]
}

```

#### 2.4.2.9 BIOS Northbridge COF and VID Configuration

BIOS is responsible for initializing the NB COF and VID settings based on the power plane capabilities of the platform. F3xD4[NbFid] must be matched between all processors in the coherent fabric of a multi-socket system. The lowest setting from all processors in a multi-socket system (determined by using the following equations on each processor and selecting the lowest value) is used as the common NbFid. The NewNbVid values derived from the following equations are applied uniquely to each processor in the system and are not matched across processors.

```

If F3x1FC[NbCofVidUpdate]=0 or F3x1FC[NbCofVidUpdate] is undefined {
    • NewNbVid = MSRC001_0071[CurNbVid]
    • NewNbFid = F3xD4[NbFid]
} else {

```

If the processor is installed in a dual-plane system:

- NewNbVid = DualPlaneNbVid (see [F3x1FC\[DualPlaneNbVidOff\]](#))
- NewNbFid = DualPlaneNbFid (see [F3x1FC\[DualPlaneNbFidOff\]](#)).

If the processor is installed in a single-plane system:

- NewNbVid = [F3x1FC\[SinglePlaneNbVid\]](#)
- NewNbFid = [F3x1FC\[SinglePlaneNbFid\]](#).

}

#### 2.4.2.9.1 BIOS NB COF and VID Configuration for SVI and Single-Plane PVI Systems

For all processors in the system, if [F3x1FC\[NbCofVidUpdate\]=0](#) or [F3x1FC\[NbCofVidUpdate\]](#) is undefined, and all processors in the system have equivalent values in [F3xD4\[NbFid\]](#), then no updates are required for the NB COF and VID configuration, and the following numbered sequence can be skipped. The sequence assumes that the processor is in the P-state specified by [MSRC001\\_0071\[StartupPstate\]](#). NewNbFid and NewNbVid are defined in section 2.4.2.9 [[BIOS Northbridge COF and VID Configuration](#)].

1. Copy the contents of the P-state register [MSRC001\\_00\[68:64\]](#) pointed to by [MSRC001\\_0071\[StartupPstate\]](#) to [MSRC001\\_0064](#) and [MSRC001\\_0065](#) for all cores on the local processor.
2. If [F3xDC\[PstateMaxVal\] == 0](#), set [F3xDC\[PstateMaxVal\]](#) to 1.
3. Copy NewNbVid to [MSRC001\\_0064\[NbVid\]](#) for all cores on the local processor.
4. Request a transition to P1 (Write [MSRC001\\_0062\[PstateCmd\]=001b](#)) for all cores on the local processor.
5. Request a transition to P0 (Write [MSRC001\\_0062\[PstateCmd\]=000b](#)) on core 0 of the local processor.
6. Wait for [MSRC001\\_0063\[CurPstate\]=000b](#) on core 0 of the local processor.
7. If [F3xDC\[PstateMaxVal\] == 0](#) at the beginning of step 2, clear [F3xDC\[PstateMaxVal\]](#).
8. Copy NewNbFid to [F3xD4\[NbFid\]](#) and set [F3xD4\[NbFidEn\]](#) on the local processor.
9. Repeat steps 1 through 7 for each processor in the system.
10. Issue a warm reset. This is required to cause the new [F3xD4\[NbFid\]](#) setting(s) to be applied and resets the values in [MSRC001\\_00\[68:64\]](#).
11. Update [MSRC001\\_00\[68:64\]\[NbVid\]](#) according to [F3x1FC\[NbVidUpdateAll\]](#) as follows:
  - If [F3x1FC\[NbVidUpdateAll\]=0](#) copy NewNbVid to [MSRC001\\_00\[68:64\]\[NbVid\]](#) where [MSRC001\\_00\[68:64\]\[NbDid\]=0](#) and [MSRC001\\_00\[68:64\]\[IddValue\]!=0](#).
  - If [F3x1FC\[NbVidUpdateAll\]=1](#) copy NewNbVid to [MSRC001\\_00\[68:64\]\[NbVid\]](#) where [MSRC001\\_00\[68:64\]\[IddValue\]!=0](#).
12. For each processor in the system, transition all cores to [MSRC001\\_0071\[StartupPstate\]](#) using [MSRC001\\_0062\[PstateCmd\]](#).

#### 2.4.2.9.2 BIOS NB COF and VID Configuration for Dual-Plane PVI Systems

If the systemboard is dual-plane and [F3xA0\[PviMode\]=1](#), then systemboard-specific control logic exists for the routing of the processor VID pins to the voltage regulators. Note that the devices used to route the processor VID[5:0] lines to the VDD and VDDNB regulators may not be transparent after cold or warm reset, but are required to be transparent when transitioning from S4 or S5 to S0. The steps in the following sequence that manipulate VID control logic on the systemboard affect all sockets. The following sequence is used to properly configure VDD and VDDNB:

1. Expose both the VDD and VDDNB regulator inputs to the VID code driven on the processor VID[5:0] lines.
2. Wait a sufficient time for the systemboard-specific control logic to pass the processor VID[5:0] value to the regulator inputs.
3. Latch the VID code driven on the processor VID[5:0] lines and preserve the value to the VDD regulator input while leaving the VDDNB regulator input exposed to transitions on the processor VID[5:0] lines.
4. Copy the contents of the P-state register [MSRC001\\_00\[68:64\]](#) pointed to by [MSRC001\\_0071\[StartupP-](#)

- state] to MSRC001\_0064 and MSRC001\_0065 for all cores on the local processor.
5. Copy NewNbVid to MSRC001\_0064[NbVid] for all cores on the local processor.
  6. If F3xDC[PstateMaxVal] == 0, set F3xDC[PstateMaxVal] to 1.
  7. Request a transition to P1 (Write MSRC001\_0062[PstateCmd]=001b) for all cores on the local processor.
  8. Request a transition to P0 (Write MSRC001\_0062[PstateCmd]=000b) on core 0 of the local processor.
  9. Wait for MSRC001\_0063[CurPstate]=000b on core 0 of the local processor.
  10. Copy NewNbFid to F3xD4[NbFid] and set F3xD4[NbFidEn] on the local processor.
  11. Repeat steps 4 through 9 for each processor in the system.
  12. Latch the VID code driven on the processor VID[5:0] lines and preserve the value to the VDDNB regulator.
  13. Request a transition to P1 (Write MSRC001\_0062[PstateCmd]=001b) on core 0 of each processor.
  14. Expose the VDD regulator inputs to the VID code driven on the processor VID[5:0] lines while leaving the VDDNB regulator input latched.
  15. Issue a warm reset. This is required to cause the new F3xD4[NbFid] setting(s) to be applied and resets the values in MSRC001\_00[68:64].
  16. Copy MSRC001\_00[68:64][CpuVid] to MSRC001\_00[68:64][NbVid] for every core in the system.
  17. For each processor in the system, transition all cores to MSRC001\_0071[StartupPstate] using MSRC001\_0062[PstateCmd].

#### 2.4.2.10 Processor-Systemboard Power Removal Check

The processor's TDP specifies the thermal solution required for proper operation. Software uses the following equation to calculate the TDP of the processor:

TDP = voltage specified by F3xD8[TdpVid] \* ProcIddMax.

ProcIddMax is calculated using the equation defined in 2.4.2.11 [Processor-Systemboard Power Delivery Compatibility Check]. If F3x1FC[DualPlaneOnly] == 1 and the algorithm in 2.4.2.7 [BIOS Configuration for Dual-plane Only Support] has not been run, ProcIddMax is calculated using MSRC001\_0068. For revision D and earlier processors, ProcIddMax is calculated using MSRC001\_0064. For revision E processors, ProcIddMax is calculated using MSRC001\_00[68:64] indexed by F4x15C[NumBoostStates].

#### 2.4.2.11 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 section 2.4.1 [Processor Power Planes And Voltage Control] for power plane definitions and configuration information. BIOS should perform this check independently for each processor node in the coherent fabric. BIOS can optionally notify the user if P-states are detected that exceed the systemboard power delivery capability. Modifications to [The P-State [4:0] Registers] MSRC001\_00[68:64] must be applied equally to all cores on the same node. This check does not guarantee functionality for all package/socket compatible processor/systemboard combinations.

MSRC001\_00[68:64][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.

Revision B or C dual-plane systems:

$$\text{ProcIddMax} = \text{MSRC001\_00[68:64][IddValue]} * 1/10^{\text{MSRC001\_00[68:64][IddDiv]} * (\text{F3xE8[CmpCap]+1}) - \text{F3x1FC[SinglePlaneNbIdd]} * 2;$$

Revision B or C single-plane systems:

$$\text{ProcIddMax} = \text{MSRC001\_00}[68:64][\text{IddValue}] * 1/10^{\text{MSRC001\_00}[68:64][\text{IddDiv}]} * (\text{F3xE8}[\text{CmpCap}] + 1);$$

Revision D and later systems:

$$\text{ProcIddMax} = \text{MSRC001\_00}[68:64][\text{IddValue}] * 1/10^{\text{MSRC001\_00}[68:64][\text{IddDiv}]} * (\text{F3xE8}[\text{CmpCap}] + 1) * (\text{F3xE8}[\text{MultiNodeCpu}] + 1);$$

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 software 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.

Revision B or C single power plane and revision D P1 example:

- $\text{MSRC001\_0065}[\text{IddValue}] = 32\text{d}$
- $\text{MSRC001\_0065}[\text{IddDiv}] = 0$
- $\text{F3xE8}[\text{CmpCap}] = 1$
- $\text{F3xE8}[\text{MultiNodeCpu}] = 0$
- $\text{ProcIddMax} = 32 * 1 * 2 * 1 = 64\text{A}$  per plane

The systemboard must be able to supply  $\geq 64\text{A}$  per plane for the unified core power plane in order to support P1 for this processor. If the systemboard current delivery capability is  $< 64\text{A}$  per plane then BIOS must set  $\text{MSRC001\_0065}[\text{PstateEn}] = 0$  for all cores on this processor node, and continue by checking P2 in the same fashion.

Revision B or C dual power plane P1 example:

- $\text{MSRC001\_0065}[\text{IddValue}] = 32\text{d}$
- $\text{MSRC001\_0065}[\text{IddDiv}] = 0$
- $\text{F3xE8}[\text{CmpCap}] = 1$
- $\text{F3x1FC}[\text{SinglePlaneNbIdd}] = 10\text{d}$
- $\text{ProcIddMax} = 32 * 1 * 2 - 20 = 44\text{A}$  per plane

The systemboard must be able to supply  $\geq 44\text{A}$  per plane for the unified core power plane in order to support P1 for this processor. If the systemboard current delivery capability is  $< 44\text{A}$  per plane then BIOS must set  $\text{MSRC001\_0065}[\text{PstateEn}] = 0$  (see [MSRC001\\_00\[68:64\]](#)) for all cores on this processor node, and continue by checking P2 in the same fashion.

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

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

1. If the P-state pointed to by  $\text{MSRC001\_0063}[\text{CurPstate}]$  is disabled by the power delivery compatibility check, then BIOS must request a transition to an enabled P-state using  $\text{MSRC001\_0062}[\text{PstateCmd}]$  and wait for  $\text{MSRC001\_0063}[\text{CurPstate}]$  to reflect the new value.
2. Copy the contents of the enabled P-state MSRs ([MSRC001\\_00\[68:64\]](#)) 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 ( $\text{PstateEn} = 0$ ). This step uses software P-state numbering. See [2.4.2.1.2.1 \[Software P-state Numbering\]](#).
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.
- For revision E systems with `CPUID Fn8000_0007[CPB]=1`, if P0 is disabled then program `F4x15C[BoostSrc]=0`. This step uses hardware P-state numbering. See 2.4.2.1.2.2 [Hardware P-state Numbering].
  - 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:
    - `F3x64[HtcPstateLimit]`
    - `F3x68[StcPstateLimit]`
    - `F3xDC[PstateMaxVal]`

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

- 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.
- If `MSRC001_0061[PstateMaxVal] != 000b` copy the contents of the P-state MSR pointed to by `MSRC001_0061[PstateMaxVal]` to the P0 MSR and set `MSRC001_00[68:64][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 2.4.2.1.2.1 [Software P-state Numbering].
- Adjust the following fields to 000b for revision D and earlier processors. For revision E processors adjust the following fields to `F4x15C[NumBoostStates]`.
  - `F3x64[HtcPstateLimit]`
  - `F3x68[StcPstateLimit]`
  - `F3xDC[PstateMaxVal]`
- For revision E systems with `CPUID Fn8000_0007[CPB]=1`, program `F4x15C[BoostSrc]=0`.

#### 2.4.2.12 Mixed-Frequency and Power P-State Configuration

Processors with different P-state CPU COFs and powers can be mixed in a system. All cores must have the same number of P-states, and all equivalent P-states must have identical ACPI CoreFreq and Power settings. The CoreFreq and Power values are derived from `MSRC001_00[68:64][PstateEn, CpuFid, CpuDid, IddDiv, and IddValue]` using the formulas described in section 2.4.2.13.2 [\_PSS (Performance Supported States)].

- If `MSRC001_00[68:64][PstateEn, CpuFid, CpuDid, IddDiv, and IddValue]` are identical for all processors, no BIOS modifications to [The P-State [4:0] Registers] `MSRC001_00[68:64]` are necessary.
- If `MSRC001_00[68:64][PstateEn, CpuFid, or CpuDid]` differs between processors, sections 2.4.2.12.2 [Mixed Frequency and Power P-State Configuration Rules] and 2.4.2.12.3 [Mixed Frequency and Power P-State Configuration Sequence] are used to determine the common set of P-states and define the required BIOS modifications to [The P-State [4:0] Registers] `MSRC001_00[68:64]`.
- If `MSRC001_00[68:64][IddDiv or IddValue]` differs between processors and `MSRC001_00[68:64][PstateEn, CpuFid, and CpuDid]` do not differ between processors, section 2.4.2.12.1 [Mixed Power P-State Configuration Sequence] defines the required BIOS modifications to [The P-State [4:0] Registers] `MSRC001_00[68:64]`.

##### 2.4.2.12.1 Mixed Power P-State Configuration Sequence

BIOS must match the `MSRC001_00[68:64][IddDiv and IddValue]` fields for each P-state across all processors using the following sequence. For each `MSRC001_00[68:64]` with `PstateEn=1`:

1. Read IddDiv and IddValue for all processors.
2. Calculate the resulting power for each processor using the formula documented in 2.4.2.13.2 [[\\_PSS \(Performance Supported States\)](#)].
3. Identify the highest power for all processors.
4. Program IddDiv and IddValue for all processors equal to the values for the processor with the highest calculated power.

#### 2.4.2.12.2 Mixed Frequency and Power P-State Configuration Rules

- Processors with only one enabled P-state (`F3xDC[PstateMaxVal]=000b`) cannot be mixed in a system with processors with more than one enabled P-state (`F3xDC[PstateMaxVal]!=000b`).
- Processors with `F3xE8[HTC Capable]=1` cannot be mixed in a system with processors with `F3xE8[HTC Capable]=0`.
- In a system where one or more cores are forced down to one P-state due to board power limitations (see 2.4.2.11 [[Processor-Systemboard Power Delivery Compatibility Check](#)]), all other cores in the system must be placed into the P-state specified by `F3x64[HtcPstateLimit]`. The transition to the HTC P-state can be done at any time during the BIOS POST routine. HTC P-states must be matched according to the guidelines specified in 2.4.2.12.3 [[Mixed Frequency and Power P-State Configuration Sequence](#)]. The HTC P-state limit should be used regardless of the `F3xE8[HTC Capable]` value. The remaining requirements in this section can be skipped.
- The maximum performance P-state (P0) CPU COF for the system is equivalent to the lowest P0 CPU COF for any processor in the system.
- The number of P-states for the system is equivalent to, or lower than, the least number of P-states for any processor in the system.
- All CPU COF calculations are rounded to the nearest 100 MHz frequency for the purposes of frequency matching.
- The CPU COF for any enabled P-state can be lowered by modifying the `MSRC001_00[68:64][CpuFid]` field from the cold reset value.
- The power for any enabled P-state can be modified by writing to `MSRC001_00[68:64][IddDiv, IddValue]`.
- P-states can be invalidated by setting `MSRC001_00[68:64][PstateEn]=0`.
- P-states that are disabled at cold reset should not be enabled.
- `MSRC001_00[68:64][CpuDid, CpuVid, NbDid, NbVid]` cold reset values are not modified by the mixed frequency P-state configuration sequence for any P-state.
- No P-state changes are allowed until all appropriate steps of the sequence are complete.

#### 2.4.2.12.3 Mixed Frequency and Power P-State Configuration Sequence

1. Verify the rules in section 2.4.2.12.2 [[Mixed Frequency and Power P-State Configuration Rules](#)] regarding `F3xDC[PstateMaxVal]` and `F3xE8[HTC Capable]` for all processors.
2. Match P0 CPU COF for all cores to the lowest P0 CPU COF value in the coherent fabric, and match P0 power for all cores to the highest P0 power value in the coherent fabric.
  - If all processors have only 1 enabled P-state, the following sequence should be performed on all cores:
    - Write the appropriate CpuFid value resulting from the matched CPU COF to `MSRC001_0064[CpuFid]`.
    - Copy `MSRC001_0064` to `MSRC001_0065`.
    - Write 001b to `F3xDC[PstatemaxVal]`.
    - Write 001b to `MSRC001_0062[PstateCmd]`.
    - Wait for `MSRC001_0071[CurCpuFid] = MSRC001_0065[CpuFid]`.
    - Write 000b to `MSRC001_0062[PstateCmd]`.
    - Wait for `MSRC001_0071[CurCpuFid] = MSRC001_0064[CpuFid]`.
    - Write 0b to `MSRC001_0065[PstateEn]`.

- Write 000b to **F3xDC**[PstateMaxVal] and exit the sequence (no further steps are required).

**Table 12: Representative mixed frequency P-state table example (step 2)**

MSR	Cold Reset				Post-Step 2			
	Processor 0	Processor 1	Processor 2	Processor 3	Processor 0	Processor 1	Processor 2	Processor 3
P0	2.5 GHz 90 W	2.7 GHz 90 W	3.2 GHz 100 W	2.5 GHz 70 W	2.5 GHz <b>100 W</b>	<b>2.5 GHz</b> <b>100 W</b>	2.5 GHz 100 W	2.5 GHz <b>100 W</b>
P1	2.2 GHz <sup>1</sup> 80 W	2.4 GHz 80 W	3.0 GHz <sup>1</sup> 90 W	2.3 GHz <sup>1</sup> 60 W	2.2 GHz <sup>1</sup> 80 W	2.4 GHz 80 W	3.0 GHz <sup>1</sup> 90 W	2.3 GHz <sup>1</sup> 60 W
P2	1.8 GHz 70 W	2.2 GHz <sup>1</sup> 70 W	2.4 GHz 80 W	1.8 GHz 50 W	1.8 GHz 70 W	2.2 GHz <sup>1</sup> 70 W	2.4 GHz 80 W	1.8 GHz 50 W
P3	1.2 GHz 60 W	1.0 GHz 60 W	1.6 GHz 70 W	500 MHz <sup>2</sup> 40 W	1.2 GHz 60 W	1.0 GHz 60 W	1.6 GHz 70 W	500 MHz <sup>2</sup> 40 W
P4	500 MHz <sup>2</sup> 50 W	600 MHz <sup>2</sup> 50 W	500 MHz <sup>2</sup> 60 W	N/A <sup>3</sup>	500 MHz <sup>2</sup> 50 W	600 MHz <sup>2</sup> 50 W	500 MHz <sup>2</sup> 60 W	N/A <sup>3</sup>

Refer to [Table 17](#) for notes.

- Match the CPU COF and power for P-states used by HTC:
  - Skip to step 4 if any processor reports **F3xE8**[HTC Capable]=0.
  - Set **F3x64**[HtcPstateLimit]=001b and **F3x68**[StcPstateLimit]=001b for processors with **F3x64**[HtcPstateLimit]=000b.
  - Identify the lowest CPU COF for all processors in the P-state pointed to by [\[The Hardware Thermal Control \(HTC\) Register\] F3x64\[HtcPstateLimit\]](#).
  - Modify the CPU COF pointed to by [\[The Hardware Thermal Control \(HTC\) Register\] F3x64\[HtcPstateLimit\]](#) to the previously identified lowest CPU COF value.
  - Identify the highest power for all processors in the P-state pointed to by [\[The Hardware Thermal Control \(HTC\) Register\] F3x64\[HtcPstateLimit\]](#).
  - Modify the power pointed to by [\[The Hardware Thermal Control \(HTC\) Register\] F3x64\[HtcPstateLimit\]](#) to the previously identified highest power value.

**Table 13: Representative mixed frequency P-state table example (step 3)**

MSR	Cold Reset				Post-Step 3			
	Processor 0	Processor 1	Processor 2	Processor 3	Processor 0	Processor 1	Processor 2	Processor 3
P0	2.5 GHz 90 W	2.7 GHz 90 W	3.2 GHz 100 W	2.5 GHz 70 W	2.5 GHz <b>100 W</b>	<b>2.5 GHz</b> <b>100 W</b>	2.5 GHz 100 W	2.5 GHz <b>100 W</b>
P1	2.2 GHz <sup>1</sup> 80 W	2.4 GHz 80 W	3.0 GHz <sup>1</sup> 90 W	2.3 GHz <sup>1</sup> 60 W	2.2 GHz <sup>1</sup> <b>90 W</b>	2.4 GHz 80 W	<b>2.2 GHz</b> <sup>1</sup> <b>90 W</b>	<b>2.2 GHz</b> <sup>1</sup> <b>90 W</b>
P2	1.8 GHz 70 W	2.2 GHz <sup>1</sup> 70 W	2.4 GHz 80 W	1.8 GHz 50 W	1.8 GHz 70 W	2.2 GHz <sup>1</sup> <b>90 W</b>	2.4 GHz 80 W	1.8 GHz 50 W
P3	1.2 GHz 60 W	1.0 GHz 60 W	1.6 GHz 70 W	500 MHz <sup>2</sup> 40 W	1.2 GHz 60 W	1.0 GHz 60 W	1.6 GHz 70 W	500 MHz <sup>2</sup> 40 W
P4	500 MHz <sup>2</sup> 50 W	600 MHz <sup>2</sup> 50 W	500 MHz <sup>2</sup> 60 W	N/A <sup>3</sup>	500 MHz <sup>2</sup> 50 W	600 MHz <sup>2</sup> 50 W	500 MHz <sup>2</sup> 60 W	N/A <sup>3</sup>

Refer to [Table 17](#) for notes.

- Match the CPU COF and power for the lowest performance P-state:



- If  $F3xDC[PstateMaxVal] = F3x64[HtcPstateLimit]$  for any processor, set  $PstateEn=0$  for all P-states greater than the P-state pointed to by  $F3x64[HtcPstateLimit]$  for all processors.
- Identify the lowest CPU COF for all processors in the P-state pointed to by  $F3xDC[PstateMaxVal]$ .
- Modify the CPU COF for all processors in the P-state pointed to by  $F3xDC[PstateMaxVal]$  to the previously identified lowest CPU COF value.
- Identify the highest power for all processors in the P-state pointed to by  $F3xDC[PstateMaxVal]$ .
- Modify the power for all processors in the P-state pointed to by  $F3xDC[PstateMaxVal]$  to the previously identified highest power value.

**Table 14: Representative mixed frequency P-state table example (step 4)**

MSR	Cold Reset				Post-Step 4			
	Processor 0	Processor 1	Processor 2	Processor 3	Processor 0	Processor 1	Processor 2	Processor 3
P0	2.5 GHz 90 W	2.7 GHz 90 W	3.2 GHz 100 W	2.5 GHz 70 W	2.5 GHz 100 W	2.5 GHz 100 W	2.5 GHz 100 W	2.5 GHz 100 W
P1	2.2 GHz <sup>1</sup> 80 W	2.4 GHz 80 W	3.0 GHz <sup>1</sup> 90 W	2.3 GHz <sup>1</sup> 60 W	2.2 GHz <sup>1</sup> 90 W	2.4 GHz 80 W	2.2 GHz <sup>1</sup> 90 W	2.2 GHz <sup>1</sup> 90 W
P2	1.8 GHz 70 W	2.2 GHz <sup>1</sup> 70 W	2.4 GHz 80 W	1.8 GHz 50 W	1.8 GHz 70 W	2.2 GHz <sup>1</sup> 90 W	2.4 GHz 80 W	1.8 GHz 50 W
P3	1.2 GHz 60 W	1.0 GHz 60 W	1.6 GHz 70 W	500 MHz <sup>2</sup> 40 W	1.2 GHz 60 W	1.0 GHz 60 W	1.6 GHz 70 W	500 MHz <sup>2</sup> 60 W
P4	500 MHz <sup>2</sup> 50 W	600 MHz <sup>2</sup> 50 W	500 MHz <sup>2</sup> 60 W	N/A <sup>3</sup>	500 MHz <sup>2</sup> 60 W	500 MHz <sup>2</sup> 60 W	500 MHz <sup>2</sup> 60 W	N/A <sup>3</sup>

Refer to [Table 17](#) for notes.

5. Modify  $F3xDC[PstateMaxVal]$  to indicate the lowest performance P-state with  $PstateEn$  set for each processor (step 4 can disable P-states pointed to by  $F3xDC[PstateMaxVal]$ ).
6. Match the CPU COF and power for upper intermediate P-states:
  - Upper intermediate P-states = P-states between (not including) P0 and  $F3x64[HtcPstateLimit]$ .
  - If  $F3x64[HtcPstateLimit] = 001b$  for any processor, set  $PstateEn=0$  for enabled upper intermediate P-states for all processors with  $F3x64[HtcPstateLimit] > 001b$  and skip the remaining actions for this numbered step.
  - Define each of the available upper intermediate P-states; for each processor concurrently evaluate the following loop; when any processor falls out of the loop (runs out of available upper intermediate P-states) all other processors have their remaining upper intermediate P-states invalidated ( $PstateEn=0$ ); for ( $i = F3x64[HtcPstateLimit]-1$ ;  $i > 0$ ;  $i--$ )
    - Identify the lowest CPU COF for P(i).
    - Identify the highest power for P(i).
    - Modify P(i) CPU COF for all processors to the previously identified lowest CPU COF value.
    - Modify P(i) power for all processors to the previously identified highest power value.

**Table 15: Representative mixed frequency P-state table example (step 6)**

MSR	Cold Reset				Post-Step 6			
	Processor 0	Processor 1	Processor 2	Processor 3	Processor 0	Processor 1	Processor 2	Processor 3
P0	2.5 GHz 90 W	2.7 GHz 90 W	3.2 GHz 100 W	2.5 GHz 70 W	2.5 GHz 100 W	2.5 GHz 100 W	2.5 GHz 100 W	2.5 GHz 100 W
P1	2.2 GHz <sup>1</sup> 80 W	2.4 GHz 80 W	3.0 GHz <sup>1</sup> 90 W	2.3 GHz <sup>1</sup> 60 W	2.2 GHz <sup>1</sup> 90 W	N/A <sup>3</sup>	2.2 GHz <sup>1</sup> 90 W	2.2 GHz <sup>1</sup> 90 W
P2	1.8 GHz 70 W	2.2 GHz <sup>1</sup> 70 W	2.4 GHz 80 W	1.8 GHz 50 W	1.8 GHz 70 W	2.2 GHz <sup>1</sup> 90 W	2.4 GHz 80 W	1.8 GHz 50 W
P3	1.2 GHz 60 W	1.0 GHz 60 W	1.6 GHz 70 W	500 MHz <sup>2</sup> 40 W	1.2 GHz 60 W	1.0 GHz 60 W	1.6 GHz 70 W	500 MHz <sup>2</sup> 60 W
P4	500 MHz <sup>2</sup> 50 W	600 MHz <sup>2</sup> 50 W	500 MHz <sup>2</sup> 60 W	N/A <sup>3</sup>	500 MHz <sup>2</sup> 60 W	500 MHz <sup>2</sup> 60 W	500 MHz <sup>2</sup> 60 W	N/A <sup>3</sup>

Refer to [Table 17](#) for notes.

Example description:

$F3x64[HtcPstateLimit] = 001b$  for processors 0, 2, and 3. Therefore, the conditions of the first bullet are satisfied and processor 1 must have P1 invalidated (remaining upper intermediate P-state). Execution skips to the next numbered step.

7. Match the CPU COF and power for lower intermediate P-states:

- Lower intermediate P-states = P-states between (not including)  $F3x64[HtcPstateLimit]$  and  $F3xDC[PstateMaxVal]$
- If  $F3xDC[PstateMaxVal] - F3x64[HtcPstateLimit] < 2$  for any processor, set  $PstateEn=0$  for enabled lower intermediate P-states for all processors with  $F3xDC[PstateMaxVal] - F3x64[HtcPstateLimit] > 1$  and skip the remaining actions for this numbered step.
- Define each of the available lower intermediate P-states; for each processor concurrently evaluate the following loop; when any processor falls out of the loop (runs out of available lower intermediate P-states) all other processors have their remaining lower intermediate P-states invalidated ( $PstateEn=0$ ); for ( $i = F3xDC[PstateMaxVal]-1$ ;  $i > F3x64[HtcPstateLimit]$ ;  $i--$ )
  - Identify the lowest CPU COF for P(i).
  - Identify the highest power P(i).
  - Modify P(i) CPU COF for all processors to the previously identified lowest CPU COF value.
  - Modify P(i) power for all processors to the previously identified highest power value.

**Table 16: Representative mixed frequency P-state table example (step 7)**

MSR	Cold Reset				Post-Step 7			
	Processor 0	Processor 1	Processor 2	Processor 3	Processor 0	Processor 1	Processor 2	Processor 3
P0	2.5 GHz 90 W	2.7 GHz 90 W	3.2 GHz 100 W	2.5 GHz 70 W	2.5 GHz 100 W	2.5 GHz 100 W	2.5 GHz 100 W	2.5 GHz 100 W
P1	2.2 GHz <sup>1</sup> 80 W	2.4 GHz 80 W	3.0 GHz <sup>1</sup> 90 W	2.3 GHz <sup>1</sup> 60 W	2.2 GHz <sup>1</sup> 90 W	N/A <sup>3</sup>	2.2 GHz <sup>1</sup> 90 W	2.2 GHz <sup>1</sup> 90 W
P2	1.8 GHz 70 W	2.2 GHz <sup>1</sup> 70 W	2.4 GHz 80 W	1.8 GHz 50 W	N/A <sup>3</sup>	2.2 GHz <sup>1</sup> 90 W	N/A <sup>3</sup>	1.0 GHz 70 W
P3	1.2 GHz 60 W	1.0 GHz 60 W	1.6 GHz 70 W	500 MHz <sup>2</sup> 40 W	1.0 GHz 70 W	1.0 GHz 70 W	1.0 GHz 70 W	500 MHz <sup>2</sup> 60 W
P4	500 MHz <sup>2</sup> 50 W	600 MHz <sup>2</sup> 50 W	500 MHz <sup>2</sup> 60 W	N/A <sup>3</sup>	500 MHz <sup>2</sup> 60 W	500 MHz <sup>2</sup> 60 W	500 MHz <sup>2</sup> 60 W	N/A <sup>3</sup>

Refer to [Table 17](#) for notes.

Example description:

$F3xDC[PstateMaxVal] - F3x64[HtcPstateLimit] > 1$  for all processors. Therefore, the conditions of the first bullet are not met and execution continues to bullet two.

Loop index *i* initializes to:

P3 for processors 0, 1, and 2

P2 for processor 3

On the first iteration of the loop processor 1 has the lowest CPU COF of 1.0 GHz for P(*i*), and processor 2 has the highest power of 70 W for P(*i*). The P(*i*) values of each processor are modified to 1.0 GHz and 70 W.

The loop index *i* is decremented for all processors.

Processor 1 fails the loop index test of  $i > F3x64[HtcPstateLimit]$  with  $i = 2$  and  $F3x64[HtcPstateLimit] = 2$ .

Processor 3 fails the loop index test of  $i > F3x64[HtcPstateLimit]$  with  $i = 1$  and  $F3x64[HtcPstateLimit] = 1$ .

P2 is invalidated for processors 0 and 2 (remaining lower intermediate P-state).

Execution skips to the next numbered step.

8. Place all cores into a valid COF and VID configuration corresponding to an enabled P-state:
  - Select an enabled P-state not equal to the P-state pointed to by `MSRC001_0063[CurPstate]` for each core.
  - Transition all cores to the selected P-states by writing the Control value from the `_PSS` object corresponding to the selected P-state to `MSRC001_0062[PstateCmd]`.
  - Wait for all cores to report the Status value from the `_PSS` object corresponding to the selected P-state in `MSRC001_0063[CurPstate]`.

**Table 17: Representative mixed frequency P-state table example (final)**

MSR	Cold Reset				Post-Algorithm			
	Processor 0	Processor 1	Processor 2	Processor 3	Processor 0	Processor 1	Processor 2	Processor 3
P0	2.5 GHz 90 W	2.7 GHz 90 W	3.2 GHz 100 W	2.5 GHz 70 W	2.5 GHz <i>100 W</i>	2.5 GHz <i>100 W</i>	2.5 GHz 100 W	2.5 GHz <i>100 W</i>
P1	2.2 GHz <sup>1</sup> 80 W	2.4 GHz 80 W	3.0 GHz <sup>1</sup> 90 W	2.3 GHz <sup>1</sup> 60 W	2.2 GHz <sup>1</sup> 90 W	N/A <sup>3</sup>	2.2 GHz <sup>1</sup> 90 W	2.2 GHz <sup>1</sup> 90 W
P2	1.8 GHz 70 W	2.2 GHz <sup>1</sup> 70 W	2.4 GHz 80 W	1.8 GHz 50 W	N/A <sup>3</sup>	2.2 GHz <sup>1</sup> 90 W	N/A <sup>3</sup>	1.0 GHz 70 W
P3	1.2 GHz 60 W	1.0 GHz 60 W	1.6 GHz 70 W	500 MHz <sup>2</sup> 40 W	1.0 GHz 70 W	1.0 GHz 70 W	1.0 GHz 70 W	500 MHz <sup>2</sup> 60 W
P4	500 MHz <sup>2</sup> 50 W	600 MHz <sup>2</sup> 50 W	500 MHz <sup>2</sup> 60 W	N/A <sup>3</sup>	500 MHz <sup>2</sup> 60 W	500 MHz <sup>2</sup> 60 W	500 MHz <sup>2</sup> 60 W	N/A <sup>3</sup>

Notes:

- 1) Indicates the P-state pointed to by [F3x64\[HtcPstateLimit\]](#).
- 2) Indicates the P-state pointed to by [F3xDC\[PstateMaxVal\]](#).
- 3) N/A indicates a P-state with [MSRC001\\_00\[68:64\]\[PstateEn\]=0](#).

*Italics* indicates values modified by the mixed frequency P-state algorithm from the cold reset value.

***Bold Italics*** indicates values modified by this step of the mixed frequency P-state algorithm from the cold reset value.

[MSRC001\\_00\[68:64\]\[CpuVid, NbVid, NbDid\]](#) are not modified by the mixed frequency P-state algorithm and are not shown.

**Table 18: Representative mixed frequency \_PSS object example**

P-state	Post-Algorithm _PSS			
	Processor 0	Processor 1	Processor 2	Processor 3
0	CoreFreq = 2.5 GHz Power = 100 W Control = Status = 0h	CoreFreq = 2.5 GHz Power = 100 W Control = Status = 0h	CoreFreq = 2.5 GHz Power = 100 W Control = Status = 0h	CoreFreq = 2.5 GHz Power = 100 W Control = Status = 0h
1	CoreFreq = 2.2 GHz* Power = 90 W Control = Status = 1h	CoreFreq = 2.2 GHz* Power = 90 W Control = Status = 2h	CoreFreq = 2.2 GHz* Power = 90 W Control = Status = 1h	CoreFreq = 2.2 GHz* Power = 90 W Control = Status = 1h
2	CoreFreq = 1.0 GHz Power = 70 W Control = Status = 3h	CoreFreq = 1.0 GHz Power = 70 W Control = Status = 3h	CoreFreq = 1.0 GHz Power = 70 W Control = Status = 3h	CoreFreq = 1.0 GHz Power = 70 W Control = Status = 2h
3	CoreFreq = 500 MHz Power = 60 W Control = Status = 4h	CoreFreq = 500 MHz Power = 60 W Control = Status = 4h	CoreFreq = 500 MHz Power = 60 W Control = Status = 4h	CoreFreq = 500 MHz Power = 60 W Control = Status = 3h

Notes:

\* Indicates the P-state pointed to by [F3x64\[HtcPstateLimit\]](#) at cold reset.

Refer to section 2.4.2.13.2 [[\\_PSS \(Performance Supported States\)](#)] for details on \_PSS object creation and field definitions for CoreFreq, Power, Control, and Status.

TransitionLatency and BusMasterLatency are not modified by the by the mixed frequency P-state algorithm and are not shown.

Units are not indicative of the conventions required by the ACPI \_PSS object. Refer to section 2.4.2.13.2 [[\\_PSS \(Performance Supported States\)](#)] for details on \_PSS object creation.

### 2.4.2.13 ACPI Processor P-State Objects

ACPI 2.0 and ACPI 3.0 processor performance control for processors reporting [CPUID Fn8000\\_0007\[HwPstate\]=1](#) is implemented through two objects whose presence indicates to the OS that the platform and CPU are capable of supporting multiple performance states. Processor performance states are not supported with ACPI 1.0b. BIOS must provide the \_PCT object, \_PSS object, and define other ACPI parameters to support operating systems that provide native support for processor P-state transitions. Other optional ACPI objects are also described in the following sections.

The following rules apply to BIOS generated ACPI objects. Refer to the appropriate ACPI specification (<http://www.acpi.info>) for additional details:

- In a multiprocessing environment, all processors must support the same number of performance states.
- Each processor performance state must have identical performance and power-consumption parameters.
- Performance objects must be present under each processor object in the system.
- In a system where one or more cores are forced down to one P-state due to board power limitations (see [2.4.2.11 \[Processor-Systemboard Power Delivery Compatibility Check\]](#)), no ACPI objects should be generated.

#### 2.4.2.13.1 \_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\[HwPstate\]=1](#).

- Perf\_Ctrl\_Register = Functional Fixed Hardware
- Perf\_Status\_Register = Functional Fixed Hardware

#### 2.4.2.13.2 \_PSS (Performance Supported States)

A unique \_PSS entry is created for each P-state. BIOS must loop through each of [[The P-State \[4:0\] Registers](#)] [MSRC001\\_00\[68:64\]](#) applying the formulas for CoreFreq and Power, and assigning Control and Status appropriately for enabled P-states (PstateEn=1). The TransitionLatency and BusMasterLatency values can be calculated once for each processor and applied to all \_PSS entries for cores on that processor.

The value contained in the Control field is written to [[The P-State Control Register](#)] [MSRC001\\_0062](#) to request a P-state change to the CoreFreq of the associated \_PSS object. The value in the Control field is a direct indication of the P-state register ([MSRC001\\_00\[68:64\]](#)) that contains the COF and VID settings for the associated P-state. The value contained in [[The P-State Status Register](#)] [MSRC001\\_0063](#) can be used to identify the \_PSS object of the current P-state by equating [MSRC001\\_0063\[CurPstate\]](#) to the value of the Status field. Refer to section 2.4.2 [[P-states](#)] for further details on P-state definition and behavior.

- CoreFreq (MHz) = Calculated using the formula for ‘CPU COF’ documented in [MSRC001\\_00\[68:64\]\[CpuFid\]](#). All CoreFreq values must be rounded to the nearest 100 MHz frequency resulting in a maximum of 50 MHz frequency difference between the reported CoreFreq and calculated CPU COF.
- Power (mW)
  - Convert [MSRC001\\_00\[68:64\]\[CpuVid\]](#) to a voltage by referring to section 2.4.1.6 [[VID Encodings](#)]
  - $\text{Power(mW)} = \text{voltage} * \text{MSRC001_00[68:64][IddValue]} * 1/10^{\text{MSRC001_00[68:64][IddDiv]}} * 1000$

- TransitionLatency (us) and BusMasterLatency (us)
  - If `MSRC001_00[68:64][CpuFid]` is the same value for all P-states where `MSRC001_00[68:64][PstateEn]=1`:  $\text{TransitionLatency} = \text{BusMasterLatency} = (12 * (\text{F3xD4}[\text{PowerStepDown}] + \text{F3xD4}[\text{PowerStepUp}]) / 1000)$  us
  - If `MSRC001_00[68:64][CpuFid]` is different for any P-states where `MSRC001_00[68:64][PstateEn]=1`:  $\text{TransitionLatency} = \text{BusMasterLatency} = (12 * (\text{F3xD4}[\text{PowerStepDown}] + \text{F3xD4}[\text{PowerStepUp}]) / 1000)$  us + `F3xA0[PIILockTime]` us  
 Example:  
`MSRC001_00[68:64][CpuFid] = 4h (2000 MHz) for P0, P3, and P4`  
`MSRC001_00[68:64][CpuFid] = 3h (1800 MHz) for P1`  
`MSRC001_00[68:64][CpuFid] = 2h (1600 MHz) for P2`  
`F3xD4[PowerStepDown] = F3xD4[PowerStepUp] = 8h (50 ns/step)`  
`F3xA0[PIILockTime] = 011b (4 us)`  
 $\text{TransitionLatency} = \text{BusMasterLatency} = (12 * (50 + 50) / 1000)$  us + 4 us = 5.2 us (round up to 6 us)
- Control
  - If `MSRC001_0064 (P0)`: Control = 0000\_0000h
  - If `MSRC001_0065 (P1)`: Control = 0000\_0001h
  - If `MSRC001_0066 (P2)`: Control = 0000\_0002h
  - If `MSRC001_0067 (P3)`: Control = 0000\_0003h
  - If `MSRC001_0068 (P4)`: Control = 0000\_0004h
- Status
  - If `MSRC001_0064 (P0)`: Status = 0000\_0000h
  - If `MSRC001_0065 (P1)`: Status = 0000\_0001h
  - If `MSRC001_0066 (P2)`: Status = 0000\_0002h
  - If `MSRC001_0067 (P3)`: Status = 0000\_0003h
  - If `MSRC001_0068 (P4)`: Status = 0000\_0004h

### 2.4.2.13.3 `_PPC (Performance Present Capabilities)`

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

### 2.4.2.13.4 `_PSD (P-State Dependency)`

For revision C and later single-link processors, the ACPI 3.0 `_PSD` object should be generated for each core as follows:

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

For all other processors, the ACPI 3.0 `_PSD` object must be generated for each core as follows:

- `NumberOfEntries = 5`.
- `Revision = 0`.
- `Domain = CPUID Fn0000_0001_EBX[LocalApicId]`.
- `CoordType = FDh. (SW_ANY)`
- `NumProcessors = 1`.

For revision C and later single-link processors, BIOS should provide an option to choose between either `_PSD`



definition.

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

BIOS must declare the following FADT entries as 0. BIOS-controlled P-state transitions, if any, must be performed near the beginning of the POST routine before control is passed to the operating system. All subsequent transitions are made by system software not the BIOS. System Management Mode is not used for P-state control.

- PSTATE\_CNT = 00h
- CST\_CNT = 00h

#### 2.4.2.14 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.

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

Warm reset is asynchronous and can interrupt P-state transitions leaving the processor in a COF and VID state not specified in [The P-State [4:0] Registers] MSRC001\_00[68:64]. Refer to section 2.4.2.5 [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 section 2.4.2.15.2 [Core Maximum P-State Transition Sequence After Warm Reset] and 2.4.2.15.3 [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 2.4.2.11 [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 F3xD4[NbFid].

##### 2.4.2.15.1 Core P-State Voltage Alignment After Warm Reset

This sequence should only be performed on systems that support CPB. See 2.4.2.1.1 [Core Performance Boost (CPB)].

1. Write MSRC001\_0063[CurPstate] to MSRC001\_0062[PstateCmd] on every core in the processor.
2. Wait for MSRC001\_0071[CurCpuVid] >= [CpuVid] from MSRC001\_00[68:64] indexed by F4x15C[NumBoostStates].

##### 2.4.2.15.2 Core Maximum P-State Transition Sequence After Warm Reset

1. Modify F3xDC[PstateMaxVal] to reflect the lowest performance P-state supported, as indicated in MSRC001\_00[68:64][PstateEn].
2. If MSRC001\_0071[CurNbDid] = 0, set MSRC001\_001F[GfxNbPstateDis].
3. If MSRC001\_0071[CurPstate] != MSRC001\_0071[CurPstateLimit], go to step 18.
4. If F3xDC[PstateMaxVal] >= MSRC001\_0071[CurPstateLimit] + 1, go to step 16.
5. If F3xDC[PstateMaxVal] != 4, go to step 7.
6. Exit the sequence (no further steps can be executed) since MSRC001\_0071[CurPstate] = MSRC001\_0071[CurPstateLimit] = F3xDC[PstateMaxVal] = 4.
7. Copy [The P-State [4:0] Registers] MSRC001\_00[68:64] pointed to by MSRC001\_0071[CurPstateLimit] to [The P-State [4:0] Registers] MSRC001\_00[68:64] pointed to by MSRC001\_0071[CurPstateLimit] + 1.
8. Write MSRC001\_0071[CurPstateLimit] + 1 to F3xDC[PstateMaxVal].

9. Write `MSRC001_0061[CurPstateLimit] + 1` to `MSRC001_0062[PstateCmd]`.
10. Wait for `MSRC001_0071[CurCpuFid] = CpuFid` from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `MSRC001_0071[CurPstateLimit] + 1` and `MSRC001_0071[CurCpuDid] = CpuDid` from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `MSRC001_0071[CurPstateLimit] + 1`.
11. Copy `MSRC001_0061[CurPstateLimit]` to `MSRC001_0062[PstateCmd]`.
12. Wait for `MSRC001_0071[CurCpuFid] = CpuFid` from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `MSRC001_0071[CurPstateLimit]` and `MSRC001_0071[CurCpuDid] = CpuDid` from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `MSRC001_0071[CurPstateLimit]`.
13. If required, transition the NB COF and VID to the NbDid and NbVid from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `MSRC001_0071[CurPstateLimit]` using the sequence defined in section 2.4.2.15.4 [NB COF and VID Transition Sequence After Warm Reset].
14. Write `MSRC001_00[68:64][PstateEn]=0` for the P-state pointed to by `F3xDC[PstateMaxVal]`.
15. Write `MSRC001_0071[CurPstateLimit]` to `F3xDC[PstateMaxVal]` and exit the sequence (no further steps are required).
16. Write `MSRC001_0061[CurPstateLimit] + 1` to `MSRC001_0062[PstateCmd]`.
17. Wait for `MSRC001_0071[CurCpuFid] = CpuFid` from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `MSRC001_0071[CurPstateLimit] + 1` and `MSRC001_0071[CurCpuDid] = CpuDid` from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `MSRC001_0071[CurPstateLimit] + 1`.
18. Copy `MSRC001_0061[CurPstateLimit]` to `MSRC001_0062[PstateCmd]`.
19. Wait for `MSRC001_0071[CurCpuFid] = CpuFid` from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `MSRC001_0071[CurPstateLimit]` and `MSRC001_0071[CurCpuDid] = CpuDid` from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `MSRC001_0071[CurPstateLimit]`.
20. Set `MSRC001_001F[GfxNbPstateDis]`.
21. Issue an LDTSTOP assertion in the IO hub and exit the sequence (no further steps are required). This is required to cause the new `F0x[E4, C4, A4, 84][WidthIn, WidthOut]` settings to be applied in cases where step 19 does not result in an NB P-state change.
22. If required, transition the NB COF and VID to the NbDid and NbVid from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `MSRC001_0071[CurPstateLimit]` using the sequence defined in section 2.4.2.15.4 [NB COF and VID Transition Sequence After Warm Reset].

### 2.4.2.15.3 Core Minimum P-State Transition Sequence After Warm Reset

1. Modify `F3xDC[PstateMaxVal]` to reflect the lowest performance P-state supported, as indicated in `MSRC001_00[68:64][PstateEn]`.
2. If `MSRC001_0071[CurNbDid] = 0`, set `MSRC001_001F[GfxNbPstateDis]`.
3. If `MSRC001_0071[CurPstate] != F3xDC[PstateMaxVal]`, go to step 20.
4. If `F3xDC[PstateMaxVal] = 0` or `F3xDC[PstateMaxVal] != 4`, go to step 7.
5. If `MSRC001_0061[CurPstateLimit] <= MSRC001_0061[PstateMaxVal] - 1`, go to step 17.
6. Exit the sequence (no further steps can be executed) since `MSRC001_0071[CurPstate] = MSRC001_0071[CurPstateLimit] = F3xDC[PstateMaxVal] = 4`.
7. Copy [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `F3xDC[PstateMaxVal]` to [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `F3xDC[PstateMaxVal] + 1`.
8. Write `F3xDC[PstateMaxVal] + 1` to `F3xDC[PstateMaxVal]`.
9. Write `MSRC001_0061[PstateMaxVal]` to `MSRC001_0062[PstateCmd]`.
10. Wait for `MSRC001_0071[CurCpuFid] = CpuFid` from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `F3xDC[PstateMaxVal]` and `MSRC001_0071[CurCpuDid] = CpuDid` from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `F3xDC[PstateMaxVal]`.
11. Copy `MSRC001_0061[PstateMaxVal] - 1` to `MSRC001_0062[PstateCmd]`.
12. Wait for `MSRC001_0071[CurCpuFid] = CpuFid` from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `F3xDC[PstateMaxVal] - 1` and `MSRC001_0071[CurCpuDid] = CpuDid` from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `F3xDC[PstateMaxVal] - 1`.



13. If `MSRC001_0071[CurNbDid] = 1`, set `MSRC001_001F[GfxNbPstateDis]`.
14. If required, transition the NB COF and VID to the NbDid and NbVid from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `F3xDC[CurPstateLimit]` using the sequence defined in section 2.4.2.15.4 [NB COF and VID Transition Sequence After Warm Reset].
15. Write `MSRC001_00[68:64][PstateEn]=0` for the P-state pointed to by `F3xDC[PstateMaxVal]`.
16. Write `F3xDC[PstateMaxVal] - 1` to `F3xDC[PstateMaxVal]` and exit the sequence (no further steps are required).
17. Copy `MSRC001_0061[PstateMaxVal] - 1` to `MSRC001_0062[PstateCmd]`.
18. Wait for `MSRC001_0071[CurCpuFid] = CpuFid` from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `F3xDC[PstateMaxVal] - 1` and `MSRC001_0071[CurCpuDid] = CpuDid` from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `F3xDC[PstateMaxVal] - 1`.
19. If `MSRC001_0071[CurNbDid] = 0`, set `MSRC001_001F[GfxNbPstateDis]`.
20. Copy `MSRC001_0061[PstateMaxVal]` to `MSRC001_0062[PstateCmd]`.
21. Wait for `MSRC001_0071[CurCpuFid] = CpuFid` from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `F3xDC[PstateMaxVal]` and `MSRC001_0071[CurCpuDid] = CpuDid` from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `F3xDC[PstateMaxVal]`.
22. If `MSRC001_0071[CurNbDid] = 1`, set `MSRC001_001F[GfxNbPstateDis]` and exit the sequence (no further steps are required).
23. Issue an LDTSTOP assertion in the IO hub and exit the sequence (no further steps are required). This is required to cause the new `F0x[E4, C4, A4, 84][WidthIn, WidthOut]` settings to be applied.
24. If required, transition the NB COF and VID to NbDid and NbVid from [The P-State [4:0] Registers] `MSRC001_00[68:64]` pointed to by `F3xDC[PstateMaxVal]` using the sequence defined in section 2.4.2.15.4 [NB COF and VID Transition Sequence After Warm Reset].

#### 2.4.2.15.4 NB COF and VID Transition Sequence After Warm Reset

If the destination NbDid=0:

1. Transition `MSRC001_0070[NbVid]` to the destination NbVid using the sequence defined in section 2.4.1.9 [Software-Initiated Voltage Transitions].
2. Write 0 to `MSRC001_0070[NbDid]`.
3. Wait for `MSRC001_0071[CurNbDid]=0`.

If the destination NbDid=1:

1. Write 1 to `MSRC001_0070[NbDid]` on all cores.
2. Wait for `MSRC001_0071[CurNbDid]=1` on all cores.
3. Transition `MSRC001_0070[NbVid]` to the destination NbVid using the sequence defined in section 2.4.1.9 [Software-Initiated Voltage Transitions].

### 2.4.3 C-states

C-states are processor power states in which the processor is powered but may or may not execute instructions. C0 is the operational state in which instructions are executed. All other C-states are low-power states in which instructions are not executed. C0 and C1 are ACPI-defined states, see the ACPI specification for details. C1E is an AMD specific state. When coming out of warm and cold reset, the processor is transitioned to the C0 state.

#### 2.4.3.1 C-state Names and Numbers

C-states are often referred to by an alphanumeric naming convention, C1, C2, C3, etc. The mapping between ACPI defined C-states and AMD specified C-state actions is not direct. See 2.4.3.2 [C-state Request Interface]. The actions taken by the processor when entering a low power C-state are specified by [The ACPI Power State Control Registers] `F3x[84:80]`.

### 2.4.3.2 C-state Request Interface

C-states are dynamically requested by software and are exposed through ACPI objects (see 2.4.3.6 [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, either by executing the HLT instruction or, for revision E, by reading from an IO address specified by `MSRC001_0073[CstateAddr]`. The processor always returns 0 for this IO read.

### 2.4.3.3 C1 Enhanced State (C1E)

The C1 enhanced state (C1E) is a stop-grant state supported by the processor. The C1E state is characterized by the following properties:

- All cores are in the halt (C1) state.
- The ACPI-defined `P_LVL3` register has been accessed.
- The chipset has issued a `STPCLK` assertion message with the appropriate SMAF for C1E entry. Note that [The ACPI Power State Control Registers] `F3x[84:80]` specify the processor clocking and voltage behavior in response to the C1E SMAF.
- The processor has issued a `STOP_GRANT` message to the chipset.

General requirements for C1E:

- The ACPI-defined C2 and C3 states must not be declared to the operating system.
- C1E should only be enabled when the platform is in ACPI power management mode.

#### 2.4.3.3.1 SMI Initiated C1E

When C1E is enabled and the processor detects that all cores have entered the halt state, the processor sends an IO write to the SMI command port in the chipset. This causes the chipset to generate an SMI. It is expected that the SMI targets all cores and therefore all cores enter SMM. The SMM handler may or may not place the system into the C1E state. See 2.4.3.3.2.1 [SMM Handler Requirements for C1E] for a description. SMI initiated C1E is only supported on single link systems.

#### 2.4.3.3.2 BIOS Requirements to Initialize SMI Initiated C1E

On all cores:

- `MSRC001_0055[SmiOnCmpHalt] = 1`.
- `MSRC001_0055[C1eOnCmpHalt] = 0`.
- `MSRC001_0055[IORd] = 0`.
- `MSRC001_0055[IOMsgAddr] =` Address of the chipset's SMI command port.
- `MSRC001_0055[IOMsgData] =` Unique number used by the SMI handler to identify this SMI source.
- `F3xD4[MTC1eEn]=0`.
- BIOS must also setup the SMM handler as described below.

##### 2.4.3.3.2.1 SMM Handler Requirements for C1E

The system may have other SMM handler functions in addition to the C1E handler. If this is the case, they may be executed before or after the C1E handler.

The SMM handler on each AP should:

```
Read a value from the SMI command port
if (value == MSRC001_0055[IOMsgData])
{
    Wait for indication from the BSC to continue
```

```

}
Resume from SMM

```

The SMM handler on the BSC should:

```

Read a value from the SMI command port
if (value == MSRC001_0055[IOMsgData])
{
  Read SMMFEC9[HLT] on all cores
  if (SMMFEC9[HLT] == 1 on all cores)
  {
    Set the BM_RLD bit (bit 1) of the ACPI-defined PM1 control register
    Read the BM_STS bit (bit 4) of the ACPI-defined PM1 status register
    if (BM_STS == 1)
    {
      Clear the BM_STS bit
      Store the value of the ACPI timer
      Issue IO read to the ACPI-defined P_LVL2 register
    }
    else
    {
      Read ACPI timer and compare to the last stored timer value
      if (time since last store value < 20ms)
      {
        Issue IO read to the ACPI-defined P_LVL2 register
      }
      else
      {
        Set the ARB_DIS bit (bit 0) of the ACPI-defined PM2 control register
        Issue an IO read to the ACPI-defined P_LVL3 register
      }
    }
  }
}
Resume from SMM

```

### 2.4.3.3.3 Hardware Initiated C1E

When C1E is enabled and the processor detects that all cores have entered the halt state, the processor sends an IO read to the ACPI-defined P\_LVL3 register. This places the system into the C1E state. Hardware initiated C1E is only supported on single link systems. Hardware initiated C1E is recommended over SMI initiated C1E for revisions where both are supported. See [Table 2](#).

#### 2.4.3.3.3.1 BIOS Requirements to Initialize Hardware Initiated C1E

On all cores:

- MSRC001\_0055[SmiOnCmpHalt] = 0.
- MSRC001\_0055[C1eOnCmpHalt] = 1.
- MSRC001\_0055[IORd] = 1.
- MSRC001\_0055[IOMsgAddr] = Address of the ACPI-defined P\_LVL3 register.
- F3xD4[MTC1eEn]=0.

#### 2.4.3.3.4 Message Triggered C1E

Message triggered C1E is only supported on G34 and C32 processors. Feature support varies by revision. See [Table 2](#). The feature is built upon dedicated hardware, a set of specific messages, a board-level sideband signal,

and timers in the IO hub.

The following board-level link-defined sideband signals are used in this protocol:

- **ALLOW\_LDTSTOP:** Board-level wired-OR signal. ALLOW\_LDTSTOP is driven by all chipset North-bridges in the system and is an input to the IO hub. When ALLOW\_LDTSTOP is asserted, the IO hub is permitted to assert LDTSTOP\_L. When ALLOW\_LDTSTOP is deasserted the IO hub must deassert LDTSTOP\_L, and keep it deasserted until ALLOW\_LDTSTOP asserts, after which a programmable period of time must pass before LDTSTOP\_L can be asserted. The IO hub must be configured to treat a deassertion of ALLOW\_LDTSTOP as also being an assertion of BMREQ#. BMREQ# is the bus-mastering request input of the IO hub, and is a distinct input from ALLOW\_LDTSTOP.
- **IDLE\_EXIT\_L:** Board-level wired-OR signal. IDLE\_EXIT\_L is driven by all processors in the system and is an input to the IO hub. IDLE\_EXIT\_L is asserted by the processor when it receives a Fixed or ExtInt interrupt while the target core is in STOP\_GRANT. When IDLE\_EXIT\_L is asserted, the IO hub initiates the C1E exit sequence. IDLE\_EXIT\_L is deasserted by the processor when the target core exits STOP\_GRANT. See [F3xA0\[IdleExitEn\]](#).

#### 2.4.3.3.5 BIOS Requirements to Initialize Message Triggered C1E

Message triggered C1E should be enabled on revision D1 and later G34 and C32 processors. To enable and configure message triggered C1E, BIOS must program the following registers on all cores and nodes in the system:

- [MSRC001\\_0055\[SmiOnCmpHalt\]](#)=0.
- [MSRC001\\_0055\[C1eOnCmpHalt\]](#)=0.
- [MSRC001\\_0055\[BmStsClrOnHltEn\]](#)=1.
- [MSRC001\\_0055\[IOMsgAddr\]](#) equals the port address of the register in the IO Hub containing the ACPI defined BM\_STS bit.
- [F3xD4\[MTC1eEn\]](#)=1.
- [F3xD4\[CacheFlushImmOnAllHalt\]](#)=1.
- [F3xD4\[StutterScrubEn\]](#)=1 if scrubbing is enabled (see [2.6.6 \[Memory Scrubbers\]](#)).
- [F3xDC\[CacheFlushOnHaltCtl\]](#) != 0.
- [MSRC001\\_0015\[HltXSpCycEn\]](#)=1 on all cores in the system.
- [F3xA0\[IdleExitEn\]](#)=1.
- [F3x188\[EnStpGntOnFlushMaskWakeup\]](#)=1.

#### 2.4.3.4 Cache Flush On Halt

A core can flush its L1 and L2 caches after it enters the Halt (C1) state. Once a core flushes its caches, probes are no longer sent to that core. This improves probing performance for cores that are not halted. Feature support varies by revision. See [Table 2](#).

When a HLT instruction is executed on a given core, a timer counts down for the amount of time specified by [F3xDC\[CacheFlushOnHaltTmr\]](#). When the timer expires, the core flushes its L1 and L2 caches to either the L3 (if the processor supports an L3) or to DRAM (if the processor does not support an L3) and the core clocks are ramped down to a divisor specified by [F3xDC\[CacheFlushOnHaltCtl\]](#). If the core exits halt for any reason, the timer resets and begins counting again upon the next execution of a HLT instruction. When the core resumes normal execution, the caches refill as normal.

### 2.4.3.5 C5

The C5 state is a stop-grant state supported by the processor. C5 is characterized by the following properties:

- The processor is in the C1E state.
- All L1 and L2 caches are flushed prior to C1E entry. See 2.4.3.4 [Cache Flush On Halt].
- The altvid voltage specified by F3xDC[AltVid] is applied. See 2.4.1.5 [Alternative Voltage (Altvid)].

The following conditions are required to support C5:

- F3xDC[CacheFlushOnHaltCtl] != 0.
- MSRC001\_0055[C1eOnCmpHalt] = 1.
- F3xDC[SlamTimeMode] = 00b or 10b.
- F3x188[EnStpGntOnFlushMaskWakeup]=1.
- F3xA0[SviHighFreqSel] = 1.

### 2.4.3.6 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 2.4.3.6.1 [\_CST].

The \_CST object is not supported with ACPI 1.0b.

#### 2.4.3.6.1 \_CST

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

- Count = 1.
- Register = MSRC001\_0073[CstateAddr].
- Type = 2.
- Latency = 75.
- Power = 0.

#### 2.4.3.6.2 \_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.

### 2.4.4 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. All registers in the processor that OS initialized must be restored or re-initialized by the OS. The method used to restore the registers is system specific.

During S3 entry, system memory enters self-refresh mode. Software is responsible for bringing memory out of self-refresh mode when resuming from S3.

The following sequence must be performed on each node. Steps 1 and 2 should only be performed once per node. When the DRAM controllers are operating in ganged mode (F2x110[DctGangEn]=1):

- Steps 3, 4, 5, 7, and 8 should only be performed on DCT0.
- Steps 6, 9 and 10 should be performed on both DCT0 and DCT1.

1. Restore [The DRAM Controller Select Low Register] F2x110.
2. Restore the following registers.
  - [The DRAM Base/Limit Registers] F1x[1, 0][7C:40]
  - [The DRAM Hole Address Register] F1xF0
  - [The DRAM Base System Address Register] F1x120
  - [The DRAM Limit System Address Register] F1x124
  - [The Swap Interleaved Region Base/Limit Register] F2x10C
  - [The DRAM Controller Select High Register] F2x114
  - [The Memory Controller Configuration Low Register] F2x118
  - [The Memory Controller Configuration High Register] F2x11C
  - [The Extended Memory Controller Configuration Low Register] F2x1B0 for revision C and later.
  - [The MCA NB Configuration Register] F3x44
  - [The Variable-Size MTRRs (MTRRphysBasen and MTRRphysMaskn)] MSR0000\_02[0F:00]
  - [The Fixed-Size MTRRs (MTRRfixn)] MSR0000\_02[6F:68, 59, 58, 50]
  - [The MTRR Default Memory Type Register (MTRRdefType)] MSR0000\_02FF
  - [The System Configuration Register (SYS\_CFG)] MSRC001\_0010
  - [The Top Of Memory Register (TOP\_MEM)] MSRC001\_001A
  - [The Top Of Memory 2 Register (TOM2)] MSRC001\_001D
  - [The Northbridge Configuration Register (NB\_CFG)] MSRC001\_001F
3. Restore the following DCT registers.
  - [The DRAM CS Base Address Registers] F2x[1, 0][5C:40]
  - [The DRAM CS Mask Registers] F2x[1, 0][6C:60]
  - [The DRAM Control Register] F2x[1, 0]78
  - [The DRAM Initialization Register] F2x[1, 0]7C
  - [The DRAM Bank Address Mapping Register] F2x[1, 0]80
  - [The DRAM MRS Register] F2x[1, 0]84
  - [The DRAM Timing Low Register] F2x[1, 0]88
  - [The DRAM Timing High Register] F2x[1, 0]8C
  - [The DRAM Configuration Low Register] F2x[1, 0]90
  - [The DRAM Controller Temperature Throttle Register] F2x[1, 0]A4 for revision D and later.
  - [The DRAM Controller Miscellaneous Register 2] F2x[1, 0]A8
4. Restore [The DRAM Configuration High Register] F2x[1, 0]94. In ungangned mode, follow the frequency initialization procedure specified in 2.8.9.8 [DRAM Channel Frequency Change].
5. Wait for F2x[1, 0]94[FreqChgInPrg]=0.
6. Restore F2x[1,0]9C\_x0D0F0[F,8:0]0[8,4,0], F2x[1, 0]9C\_x00, F2x9C\_x0A, and F2x[1, 0]9C\_x0C.
7. Restore F2x[1, 0]9C\_x04.
8. Set F2x[1, 0]90[ExitSelfRef]. For revision C and earlier processors, see 2.8.9.8 for additional requirements.
9. Wait for F2x[1, 0]90[ExitSelfRef]=0. Restore F2x[1,0]9C\_x0D0FE006 and F2x[1,0]9C\_x0D0FE007, and MSRC001\_001F[GfxNbPstateDis].
10. Restore the following registers before any accesses to DRAM are made:
  - [The DRAM DQS Receiver Enable Timing Control Registers] F2x[1, 0]9C\_x[2B:10]
  - [The DRAM Write Data Timing [High:Low] Registers] F2x[1, 0]9C\_x[3:0]0[2:1]
  - [The DRAM Write ECC Timing Register] F2x[1, 0]9C\_x[3:0]03
  - [The DRAM Read DQS Timing Control [High:Low] Registers] F2x[1, 0]9C\_x[3:0]0[6:5]
  - [The DRAM Read DQS ECC Timing Control Register] F2x[1, 0]9C\_x[3:0]07
  - [The DRAM Phy Predriver Calibration Register] F2x9C\_x0A
  - [The DRAM Phy DLL Control Register] F2x[1, 0]9C\_x0D
  - [The Data Byte DLL Configuration Register] F2x[1,0]9C\_x0D0F0[F:0]13



- [The Data Byte Configuration Register] F2x[1,0]9C\_x0D0F0[F:0]30
  - [The Clock Configuration] F2x[1,0]9C\_x0D0F2[F:0]30
  - [The C/A Pad Configuration] F2x[1,0]9C\_x0D0F[9:8][F:0]0[8,4,0]
  - [The C/A Tristate Control] F2x[1,0]9C\_x0D0F812F
11. For DDR3, restore the following registers before any accesses to DRAM are made:
- [The DRAM DQS Write Timing Control Registers] F2x[1, 0]9C\_x[4A:30]

Note: BIOS must not attempt to restore registers that are not implemented in the processor revision. See the register definition to determine what revision the register is implemented in.

Many of the systemboard power planes for the processor are powered down during S3. Refer to section 2.4.1 [Processor Power Planes And Voltage Control] for power plane descriptions. Refer to the EDS for S3 processor power plane sequencing requirements and system signal states for both inputs (e.g. PWROK, RESET\_L, and LDTSTOP\_L) and outputs (e.g. VID[\*], PSI\_L, THERMTRIP\_L, etc.) during S3. Refer to the HyperTransport™ link specification for signal sequencing requirements for PWROK, RESET\_L, and LDTSTOP\_L during S3 entry and exit, and system management message sequencing for S3 entry and exit.

### 2.4.5 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, STC, SBI, or CPB.

The following procedure calculates effective frequency using [The Max Performance Frequency Clock Count (MPERF)] MSR0000\_00E7 and [The Actual Performance Frequency Clock Count (APERF)] MSR0000\_00E8:

1. At some point in time, write 0 to both MSRs.
2. At some later point in time, read both MSRs.
3. Effective frequency = (value read from MSR0000\_00E8 / value read from MSR0000\_00E7) \* P0 frequency using software P-state numbering.

Additional notes:

- The amount of time that elapses between steps 1 and 2 is determined by software.
- It is software's responsibility to disable interrupts or any other events that may occur in between the write of MSR0000\_00E7 and the write of MSR0000\_00E8 in step 1 or between the read of MSR0000\_00E7 and the read of MSR0000\_00E8 in step 2.
- The behavior of MSR0000\_00E7 and MSR0000\_00E8 may be modified by MSRC001\_0015[EffFreqCntMwait].
- The effective frequency interface provides +/- 50MHz accuracy if the following constraints are met:
  - Effective frequency is read 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.
- MSRC001\_0015[TscFreqSel] must be set to 1 for the effective frequency interface to function correctly.
- MSR0000\_00E7 and MSR0000\_00E8 are invalid if an overflow occurs.



## 2.5 Processor State Transition Sequences

### 2.5.1 ACPI Power State Transitions

This section specifies ACPI power state transitions as controlled by the [The ACPI Power State Control Registers] F3x[84:80].

The following describes the state transition behavior associated with ACPI power state transitions:

- All SMAF controllable parameters take effect after an LDTSTOP assertion except CpuDid, which takes effect before the LDTSTOP assertion.
- CpuDid:
  - F3x[84:80][CpuDid] is applied after the processor has transitioned from C0 to a low-power state (halt or stop-grant) and the hysteresis time (F3xD4[ClkRampHystSel]) has elapsed.
  - F3x[84:80][CpuDid] is removed when the processor transitions from a low-power state to C0 or when a probe occurs and F3x[84:80][CpuPrbEn]=0.
- Probes:
  - F3x[84:80][CpuPrbEn] specifies how probes are handled while in the low-power state.
- DRAM:
  - DRAM self refresh is enabled if LDTSTOP is asserted.
  - DRAM memory clock is tristated if any of the following are true:
    - DRAM is in self refresh and the SMAF code indicates DRAM memory clock tristate can be applied (F3x[84:80][NbGateEn]=1).
    - DRAM is in self refresh and DDR3 registered DIMMs are used.
  - When LDTSTOP is deasserted the DRAM memory clock is enabled and self refresh is disabled.
    - Occurs in parallel to re-connecting the link.

Refer to the HyperTransport™ link specification for system management sequencing requirements when performing ACPI state transitions.

## 2.6 The Northbridge (NB)

Each node includes a single Northbridge that provides the interface to the local core(s), the interface to system memory, the interface to other processors, and the interface to system IO devices. The NB includes all power planes except VDD; see 2.4.1 [Processor Power Planes And Voltage Control] for more information.

The NB of each node is responsible for routing transactions sourced from cores and links to the appropriate core, cache, DRAM, or link. See 2.9.3 [Access Type Determination] for more information.

### 2.6.1 Northbridge (NB) Architecture

Major NB blocks are: System Request Interface (SRI), Memory Controller (MCT), DRAM Controllers (DCTs), L3 cache, 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 links.

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 [The DRAM CS Base Address Registers] F2x[1, 0][5C:40]. Normalized addresses include only address bits within the DCTs' range. The normalized address varies based on DCT interleave and hoisting settings in [The DRAM Controller Select Low Register] F2x110 and [The DRAM Controller Select High Register] F2x114 as well as node interleaving based on [The DRAM Base/Limit Registers] F1x[1, 0][7C:40].

## 2.6.2 The GART

The GART is a device that translates a range of physical address space, called the GART aperture, to a logical address based on page tables in system memory. The GART also includes a cache for the page table translations. The registers that specify GART behavior are:

- [The GART Aperture Control Register] F3x90.
- [The GART Aperture Base Register] F3x94.
- [The GART Table Base Register] F3x98.
- [The GART Cache Control Register] F3x9C.

Note: The GART registers must be programmed to the same value for all nodes in the system.

## 2.6.3 DMA Exclusion Vectors (DEV)

The DEV is a set of protection tables in system memory that inhibit IO accesses to ranges of system memory. The tables specify link-defined UnitIDs that are allowed access to physical memory space on a 4 KByte page basis. Multiple protection domains are supported, each with independent DEV tables and supported UnitIDs. See [The DEV Capability Header Register] F3xF0 for more details.

## 2.6.4 Northbridge Routing

There are two types of routing the NB performs to determine where to route a transaction: (1) address space routing determines which node the transaction is routed to, and (2) HyperTransport™ transaction routing determines the path in the coherent fabric that the transaction follows to reach its destination.

### 2.6.4.1 Address Space Routing

There are four main types of address space routed by the NB: (1) memory space targeting system DRAM, (2) memory space targeting IO (MMIO), (3) IO space, and (4) configuration space. The NB includes two sets of routing registers for each of these:

- Base map registers accessed through function 1, offsets 40 through F4. These are normally adequate for smaller systems.
- Extended map registers, accessed through [The Extended Address Map Data Port] F1x114. These may be needed to support larger systems.

There are no restrictions which, or both, of these map registers are enabled. If both are enabled, then the base map registers take precedence over the extended map registers.

#### 2.6.4.1.1 DRAM and MMIO Memory Space

For memory-space transactions, the physical address, cacheability type, access type, and DRAM/MMIO destination type (as specified in section 2.9.3.1.2 [Determining The Access Destination for CPU Accesses]) are presented to the NB for further processing as follows:

- Regardless of the access DRAM/MMIO destination, if supplied, the physical address is checked against the NB's AGP-aperture range-registers F3x90 and F3x94, if enabled; if the address matches, the NB translates the physical address through the AGP GART. A match in the AGP aperture overrides any match to [The DRAM Base/Limit Registers] F1x[1, 0][7C:40], and [The Memory Mapped IO Base/Limit Registers] F1x[BC:80].
  - For accesses from IO devices, the cacheability attribute from the GART entry's "Coherent" bit, as specified in [The GART Table Base Register] F3x98[GartTb1BaseAddr], is applied.
  - For accesses from a CPU, the attribute already applied by the core is used and the "Coherent" bit is ignored. (System software should ensure that the cacheability attribute assigned to an AGP aperture

matches the “Coherent” bit in the matching GART entry.)

- IO-device accesses that do not match the AGP aperture and post-GART translated addresses are compared against:
  - If the access matches [The Memory Mapped IO Base/Limit Registers] F1x[BC:80], then the transaction is routed to the specified link;
  - Else, if the access matches [The Extended MMIO Address Base Registers] F1x114\_x2 and [The Extended MMIO Address Mask Registers] F1x114\_x3, then the access is routed to the specified link;
  - Else, if the access matches [The DRAM Base/Limit Registers] F1x[1, 0][7C:40], then the access is routed to the specified link or DCT;
  - Else, the access is routed to the node or link that contains compatibility (subtractive) address space, specified by [The Node ID Register] F0x60[SbNode] and [The Unit ID Register] F0x64[SbLink].
- For core accesses that do not match the AGP aperture, the routing is determined based on the DRAM/MMIO destination:
  - If the destination is DRAM:
    - If the access matches [The DRAM Base/Limit Registers] F1x[1, 0][7C:40], then the transaction is routed to the specified link;
    - Else, the access is routed to the node or link that contains compatibility (subtractive) address space, specified by [The Node ID Register] F0x60[SbNode] and [The Unit ID Register] F0x64[SbLink].
  - If the destination is MMIO:
    - If the access matches [The Memory Mapped IO Base/Limit Registers] F1x[BC:80], then the transaction is routed to the specified link;
    - Else, if the access matches [The Extended MMIO Address Base Registers] F1x114\_x2 and [The Extended MMIO Address Mask Registers] F1x114\_x3, then the access is routed to the specified link;
    - Else, the access is routed to the node or link that contains compatibility (subtractive) address space, specified by [The Node ID Register] F0x60[SbNode] and [The Unit ID Register] F0x64[SbLink].

#### 2.6.4.1.2 IO Space

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

- If the access matches [The IO-Space Base/Limit Registers] F1x[DC:C0], then the transaction is routed to the specified link;
- Else, the access is routed to the node or link that contains compatibility (subtractive) address space, specified by [The Node ID Register] F0x60[SbNode] and [The Unit ID Register] F0x64[SbLink].

#### 2.6.4.1.3 Configuration Space

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

- If the access targets the configuration space of an existing node (based on the configuration-space address and F0x60[NodeCnt]), then it is routed to that node.
- Else, if the access matches [The Configuration Map Registers] F1x[EC:E0], then the transaction is routed to the specified link;
- Else, the access is routed to the node or link that contains compatibility (subtractive) address space, specified by [The Node ID Register] F0x60[SbNode] and [The Unit ID Register] F0x64[SbLink].

#### 2.6.4.2 HyperTransport™ Technology Routing

There are three types of HyperTransport™ transactions routed by the NB: (1) broadcast transactions, (2) request transactions, and (3) response transactions. The NB includes routing registers for each node that specify the link to route each transaction type accessed through [The Routing Table Registers] F0x[5C:40].

Figure 5 through Figure 7 show the supported topologies for G34 processors. The links between internal nodes are described in 2.7.1.5 [Link Mapping for Dual-Node Processors in the G34 Package]. The IO Hub must be connected to internal node 0.

Figure 5 shows the supported two processor topology. For the link connections between the two processors, the thick lines represent 16-bit links and the thin lines represent 8-bit links. Optionally, the 16-bit links may be swapped with the 8 bit links.

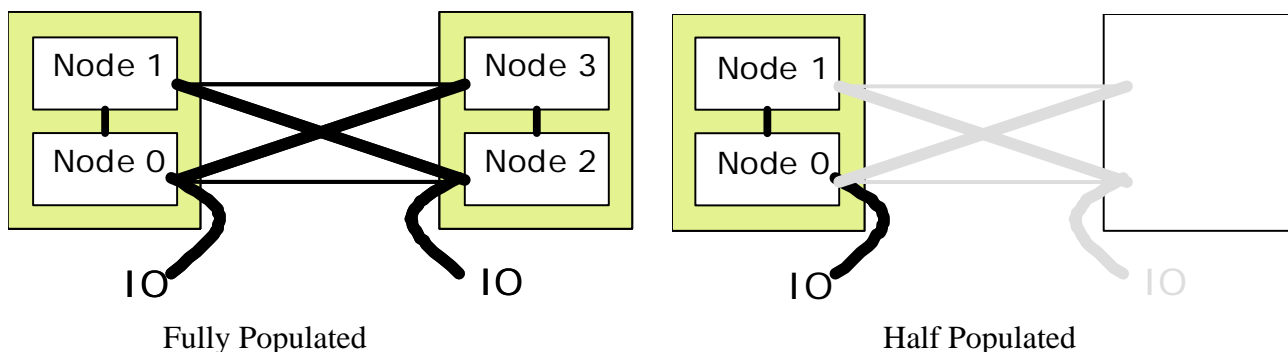


Figure 5: Two processor G34 topology

Figure 6 shows the supported four processor topology for G34 processors with two 16-bit IO links. This topology provides maximum performance. Figure 7 shows the supported four processor topology for G34 processors with four 16-bit IO links. This topology provides maximum IO. The links between processors are all 8-bits.

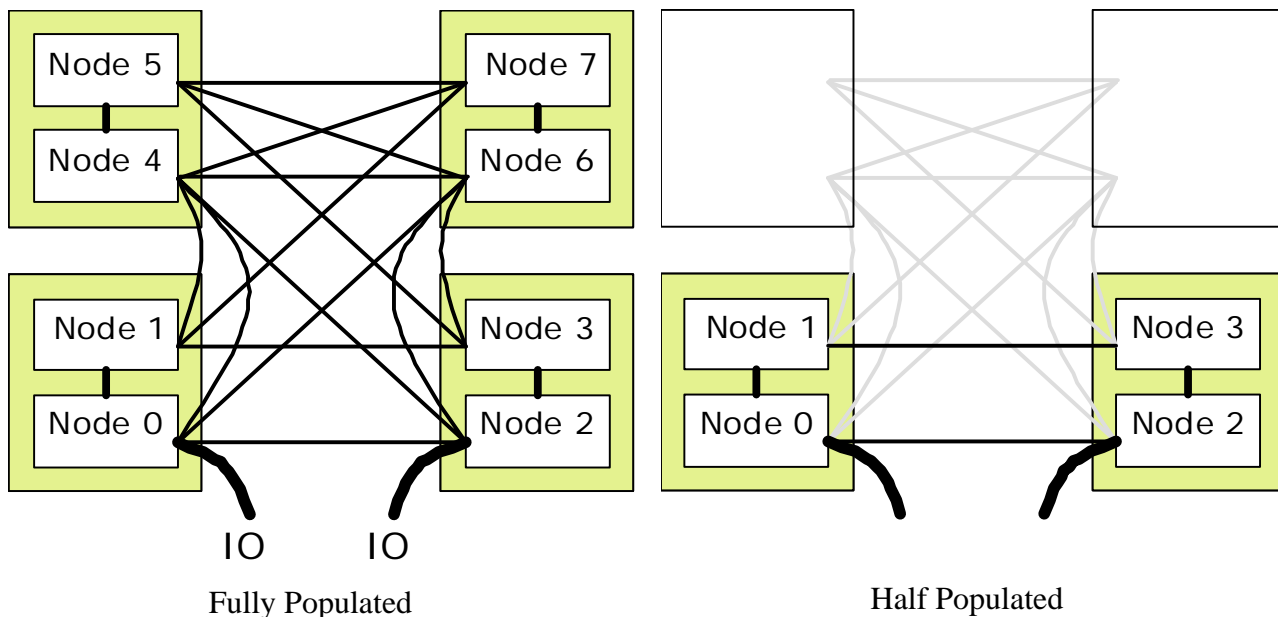
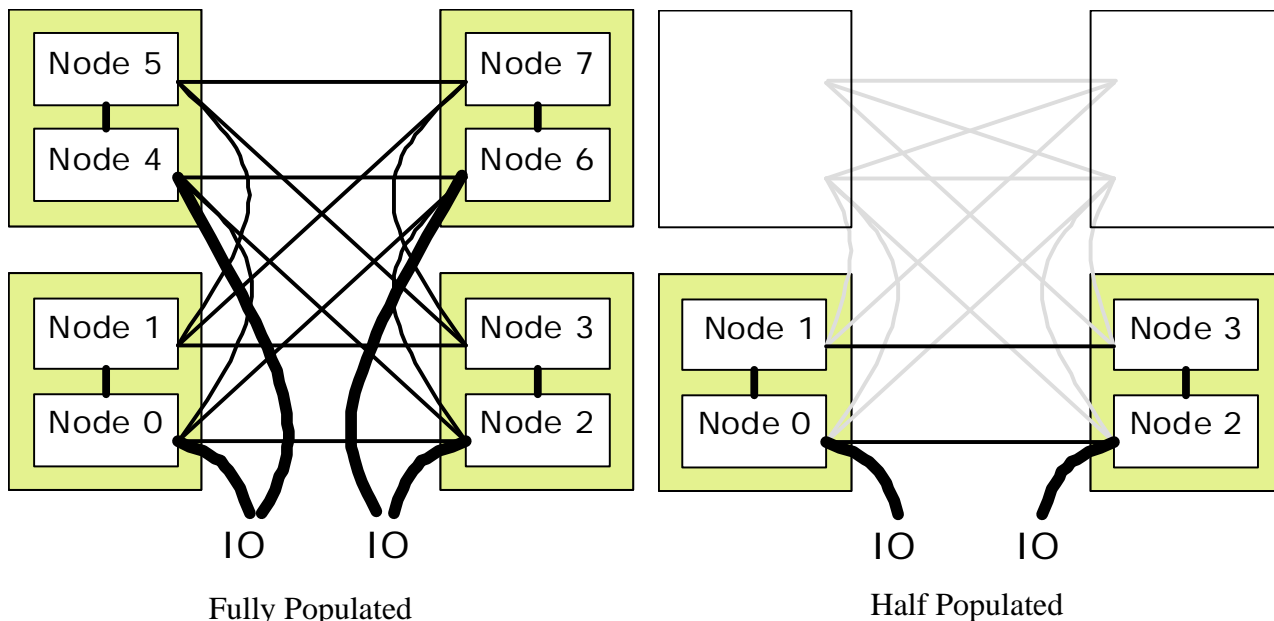


Figure 6: Four processor G34 maximum performance topology



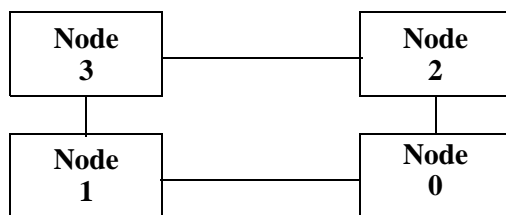
**Figure 7: Four processor G34 maximum IO topology**

**2.6.4.2.1 Routing Table Configuration**

The routing table registers must be configured correctly in multi-node systems to ensure that probes are only delivered once to each node and to ensure that the routing table is deadlock free.

A routing table is deadlock free if it contains no open-paths and no two-hop cycles.

An open-path is a routing path between nodes that traverse one or more nodes that contains a subpath that is not a routing path in the routing table. For example if the routing path between nodes 0 and 2 in Figure 8 was Node 0->Node 1->Node 3->Node 2 and the routing path between Nodes 3 and 2 was not Node 3->Node 2 then the routing path between Nodes 0 and 2 would be open because the subpath Node 3->Node 2 is not a path in the routing table.



**Figure 8: Sample four-node configuration**

A two-hop cycle is a group of two hop routing paths (routing paths between two nodes that pass through a third node) such that the first and second nodes in the each two hop routing path are also the second and third nodes in a two hop routing path in the group.

Consider the four node configuration shown in Figure 8. A two-hop cycle would occur in this configuration if the routing table was configured with the following routing paths:

- The routing path from Node 0 to Node 3 is: Node 0->Node 1->Node 3.

- The routing path from Node 1 to Node 2 is: Node 1->Node 3->Node 2.
- The routing path from Node 2 to Node 1 is: Node 2->Node 0->Node 1.
- The routing path from Node 3 to Node 0 is: Node 3->Node 2->Node 0.

To break this cycle at least one but no more than three of these routing paths must be modified to use a different intermediate node. Reconfiguring the routing paths as follows eliminates the 2-hop cycle.

- The routing path from Node 0 to Node 3 is: Node 0->Node 1->Node 3.
- The routing path from Node 1 to Node 2 is: Node 1->Node 3->Node 2.
- The routing path from Node 2 to Node 1 is: Node 2->Node 0->Node 1.
- The routing path from Node 3 to Node 0 is: Node 3->Node 1->Node 0.

#### 2.6.4.2.2 NodeId Enumeration Requirements for Dual-node Processors

The assignment of **F0x60**[NodeId] in a dual-node processor system, identified by **F3xE8**[MultiNodeCpu], is done starting with the node containing the BSC. The BSC node by definition has NodeId=0. The other internal node in that dual-node processor is assigned NodeId=1. See **F3xE8**[IntNodeNum] for how to identify the internal nodes in a dual-node processor. The remaining NodeId's in the system must be assigned using the following rules:

- Internal nodes must be assigned NodeId's in even/odd pairs. A pair of NodeId's consists of N and N+1 where N is an even integer.
- NodeId's must be assigned contiguously. For example, node ID assignment {0, 1, 2, 3} is allowed but {0, 1, 4, 5} is not.

For example, consider a 3 processor system consisting of dual-node processors A, B, and C where internal node 1 of processor B contains the BSC. A valid NodeId assignment would be:

NodeId	Processor	<b>F3xE8</b> [IntNodeNum]
0	B	0
1	B	1
2	A	0
3	A	1
4	C	1
5	C	0

#### 2.6.4.2.3 BIOS Requirements for Systems with Mixed Processor Families

Processors that are not Family 10h processor are not supported on coherent links by Family 10h processors. BIOS must ensure that all nodes in the coherent fabric are Family 10h processors by reading [\[The CPUID Family/Model Register\] F3xFC](#) before initializing the node. If a node that is not a Family 10h processor is discovered, BIOS must configure the BSP routing tables as a single processor system.

The BIOS may continue the boot process in order to display an error message on the screen if the BSP has DRAM attached and the display adapter is connected to an IO link accessible to the BSP. If these conditions are not met the BIOS may signal an error in a implementation specific manner. The BIOS must not continue the boot process after the error has been reported.

#### 2.6.4.2.4 Link Traffic Distribution

Link traffic distribution is a mechanism to reduce coherent link congestion by distributing the traffic over multiple links. For revision D and later revisions, there are two supported modes for link traffic distribution. BIOS

must ensure that only one mode is enabled in a system.

#### 2.6.4.2.4.1 Coherent Link Traffic Distribution

This mode of link traffic distribution supports 2-node systems in which multiple coherent links are connected between the nodes. For example, a 2-node system may connect 2 or 3 coherent links between the two nodes in order to increase bandwidth between them. Note: all links connected between the two nodes should be the same width (either 16-bit ganged links or 8-bit unganged sublinks). The mode is enabled by [The Coherent Link Traffic Distribution Register] F0x164. The following requirement must be met:

- For any virtual channels that are enabled for distribution, the corresponding routing table entry in F0x[5C:40] is required to select one of the links specified for distribution in F0x164[DstLnk].

#### 2.6.4.2.4.2 Coherent Link Pair Traffic Distribution

For revision D and later revisions, this mode of link traffic distribution supports up to four coherent link groupings among two or more nodes in a system. This mode is enabled by [The Coherent Link Pair Traffic Distribution Register] F0x1E0. Each link group consists of a pair of coherent links between two nodes. One of the links in the link pair is designated as the master link and the other link is designated as the alternate link. The coherent link pair can be either symmetric (same width) or asymmetric (different width). The following requirements must be met:

- For an asymmetric link pair, the master link must be the larger link.
- The master link is specified and the alternate link is not specified in F0x[5C:40].

#### 2.6.4.2.5 Display Refresh And IFCM

*Display refresh* traffic is traffic generated by UMA graphics chipsets. It targets system memory for the purpose of refreshing the display. Feature support varies by revision. See Table 2. Link display refresh packets are defined as follows:

1. IO-initiated, non-posted read requests with the isochronous bit, PassPW bit, and RespPassPW bit set, and the coherent bit cleared. The SeqID must be zero and the request must be addressed outside the GART aperture.
2. The corresponding response to these requests.

The NB prioritizes these packets such that display refresh latency and bandwidth goals may be met. To support display refresh traffic, [The Link Transaction Control Register] F0x68[DispRefModeEn] is set.

Alternatively, if supported by the chipset, link-defined isochronous flow control mode (IFCM) may be employed. IFCM is enabled through [The Link Control Registers] F0x[E4, C4, A4, 84][IsocEn]. If this bit is set for any link, then F0x68[DispRefModeEn] must be clear.

- The processor does not support peer-to-peer accesses in isochronous virtual channels. Upstream isochronous requests that target IO space are passed to the IO device in the base channel (the Isoc bit in the request packet is low); however, the Isoc bit in the downstream response to the requester is still set in such a case.
- In non-IFCM, the link-defined Isoc bit in the request packet is cleared as it is reflected downstream in a peer-to-peer access as well.

See [The Link Base Channel Buffer Count Registers] F0x[F0, D0, B0, 90], [The Data Buffer Count Register] F3x6C, [The SRI to XBAR Command Buffer Count Register] F3x70, and [The XBAR to SRI Command Buffer Count Register] F3x74 for IFCM buffer requirements and [The SRI to XCS Token Count Register] F3x140 for IFCM and display refresh token requirements.



IOMMU uses the isochronous flow control channel. If an IOMMU is present, IFCM must be enabled for all links.

### 2.6.4.2.5.1 Recommended Buffer Count Settings for Single Link Processors

The tables below show recommended buffer settings for single link processors. The following abbreviations are defined for these tables.

- NFCM is  $F0x[E4, C4, A4, 84][IsocEn]=0$  and  $F0x68[DispRefModeEn]=0$ .
- UMA DR is  $F0x68[DispRefModeEn]=1$ .
- UMA IFCM is  $F0x68[DispRefModeEn]=0$ ,  $F0x[E4, C4, A4, 84][IsocEn]=1$ , and  $F3x158[LnkToXcsDRToken]!=0$ .
- Non-UMA IFCM is  $F0x68[DispRefModeEn]=0$ ,  $F0x[E4, C4, A4, 84][IsocEn]=1$ , and  $F3x158[LnkToXcsDRToken]=0$ .

All values are in decimal. 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 19:  $F0x[F0, D0, B0, 90]$  and  $F0x[F4, D4, B4, 94]$  Recommended Settings**

Register/Field	NFCM	UMA DR	UMA IFCM	Non-UMA IFCM	
F0x90	FreeData	2	2	0	
	FreeCmd	8	8	8	
	RspData	1	1	1	
	NpReqData	1	1	1	
	ProbeCmd	0	0	0	
	RspCmd	2	2	2	
	Preq	4	4	3	6
	NpReqCmd	18	18	11	15
F0x94	IsocRspData	0	0	0	
	IsocNpReqData	0	0	0	
	IsocRspCmd	0	0	0	
	IsocPReq	0	0	1	0
	IsocNpReqCmd	0	0	7	1

**Table 20:  $F3x6C$  and  $F3x70$  Recommended Settings**

Register/Field	NFCM	UMA DR	UMA IFCM	Non-UMA IFCM	
F3x6C	IsocRspDBC	0	6	6	1
	UpRspDBC	1	1	1	1
	DatBuf24	1	1	1	1
	DnRspDBC	1	1	1	1
	DnReqDBC	1	1	1	1
	UpReqDBC	2	1	1	2

**Table 20: F3x6C and F3x70 Recommended Settings**

Register/Field	NFCM	UMA DR	UMA IFCM	Non-UMA IFCM	
F3x70	IsocRspCBC	0	6	6	1
	IsocPreqCBC	0	1	1	1
	IsocReqCBC	0	2	2	1
	UpRspCBC	4	2	2	4
	DnPreqCBC	1	1	1	1
	UpPreqCBC	1	1	1	1
	DnRspCBC	1	1	1	1
	DnReqCBC	1	1	1	1
	UpReqCBC	3	1	1	3

**Table 22: F2x118, F3x74, and F3x7C Recommended Settings**

Register/Field	NFCM	UMA DR or IFCM		Non-UMA IFCM	
F2x118[MctPriIsoc]	x	11b	10b	x	
F2x118[MctVarPriCntLmt]	0	1	0	0	
F3x74	DRReqCBC	0	12 <sup>2</sup> /15 <sup>3</sup>	9 <sup>2</sup> /15 <sup>3</sup>	0
	IsocPreqCBC	0	1	1	1
	IsocReqCBC	0	1	1	8
	ProbeCBC	8	8	8	8
	DnPreqCBC	1	0	0	0
	UpPreqCBC	1	1	1	1
	DnReqCBC	1	0	0	0
	UpReqCBC	1	1	1	1
F3x7C	Sri2XbarFreeRspDBC	0	0	0	0
	Sri2XbarFreeXreqDBC	9	7	7	7
	Sri2XbarFreeRspCBC	0	0	0	0
	Sri2XbarFreeXreqCBC	9	8	8	8
	Xbar2SriFreeListCBC <sup>1</sup>	20	12 <sup>2</sup> /9 <sup>3</sup>	15 <sup>2</sup> /9 <sup>3</sup>	17 <sup>4</sup>

1. If the processor does not include an L3 cache, this field should be reduced to satisfy the requirement of total number of XBAR to SRI command buffers less than or equal to 32. See [F3x74](#).
2. If 32 byte display refresh requests are not generated by the graphics engine. This value should be reduced by 1 for 5 core processors and reduced by 2 for 6 core processors.
3. If 32 byte display refresh requests are generated by the graphics engine. This value should be reduced by 1 for 5 core processors and reduced by 2 for 6 core processors.
4. This value should be reduced by 1 for 5 core processors and reduced by 2 for 6 core processors.

**Table 23: F3x140, F3x144, F3x1[54, 50, 4C, 48], and F3x158 Recommended Settings**

Register/Field		NFCM	UMA DR	UMA IFCM	Non-UMA IFCM
F3x140	FreeTok	8	12	12	12
	IsocRspTok	0	3	3	3
	IsocPreqTok	0	1	1	1
	IsocReqTok	0	3	3	3
	DnRspTok	1	1	1	1
	UpRspTok	3	3	3	2
	DnPreqTok	1	1	1	1
	UpPreqTok	1	1	1	1
	DnReqTok	1	1	1	1
	UpReqTok	2	2	2	2
F3x144	ProbeTok	3	3	3	3
	RspTok	3	6	6	6
F3x1[54, 50, 4C, 48]	IsocRspTok0	0	0	0	0
	IsocPreqTok0	0	1	1	0
	IsocReqTok0	0	1	1	0
	ProbeTok0	2	0	0	2
	RspTok0	2	2	2	2
	PReqTok0	2	2	2	2
	ReqTok0	2	2	2	2
	FreeTok	3	8	8	3
F3x158[LnkToXcsDrToken]	0	3	3	0	

### 2.6.4.2.5.2 Recommended Buffer Count Settings for Multi-link Processors

The following tables provide recommended buffer count settings for processors in multi-link processors. All values are in decimal.

**Table 24: F0x[F0, D0, B0, 90] and F0x[F4, D4, B4, 94] Recommended Settings**

Register/Field		NFCM IO Link	NFCM Coherent Link	IOMMU IO Link	IOMMU Coherent Link
F0x90	FreeData	$2^1/0^2$	2	0	0
	FreeCmd	8	8	8	8
	RspData	1	2	1	3
	NpReqData	1	2	1	2
	ProbeCmd	0	9	0	8
	RspCmd	2	9	2	9
	Preq	$4^1/6^2$	2	6	2
	NpReqCmd	$18^1/16^2$	4	15	4

**Table 24: F0x[F0, D0, B0, 90] and F0x[F4, D4, B4, 94] Recommended Settings**

Register/Field	NFCM IO Link	NFCM Coherent Link	IOMMU IO Link	IOMMU Coherent Link
F0x94	IsocRspData	0	0	0
	IsocNpReqData	0	0	1
	IsocRspCmd	0	0	0
	IsocPReq	0	0	0
	IsocNpReqCmd	0	0	1
1. For revision C and earlier. 2. For revision D and later.				

**Table 25: F3x6C, F3x70, F3x74, and F3x7C Recommended Settings**

Register/Field	NFCM	IOMMU	
F3x6C	IsocRspDBC	0	1
	UpRspDBC	1 <sup>1</sup> /2 <sup>2</sup>	1
	DatBuf24	1	1
	DnRspDBC	1	1
	DnReqDBC	1	1
	UpReqDBC	2	2
F3x70	IsocRspCBC	0	1
	IsocPreqCBC	0	0
	IsocReqCBC	0	1
	UpRspCBC	4 <sup>1</sup> /5 <sup>2</sup>	5
	DnPreqCBC	1	1
	UpPreqCBC	1	1
	DnRspCBC	1	1
	DnReqCBC	1	1
	UpReqCBC	3	3
F3x74	DRReqCBC	0	0
	IsocPreqCBC	0	0
	IsocReqCBC	0	1
	ProbeCBC	8	8
	DnPreqCBC	1	1
	UpPreqCBC	1	1
	DnReqCBC	1	1
	UpReqCBC	1	1

**Table 25: F3x6C, F3x70, F3x74, and F3x7C Recommended Settings**

Register/Field		NFCM	IOMMU
F3x7C	Sri2XbarFreeRspDBC	0	0
	Sri2XbarFreeXreqDBC	9	9
	Sri2XbarFreeRspCBC	0	0
	Sri2XbarFreeXreqCBC	9 <sup>1</sup> /10 <sup>2</sup>	9
	Xbar2SriFreeListCBC	20 <sup>1</sup> /22 <sup>3</sup> /24 <sup>4</sup>	23 <sup>1</sup> /21 <sup>3</sup> /23 <sup>4</sup>
1. For revision C and earlier. 2. For revision D and later. 3. For revision D and later with 6 enabled cores per node. 4. For revision D and later with 4 enabled cores per node.			

**Table 26: F3x140, F3x144, and F3x1[54, 50, 4C, 48] Recommended Settings for Socket F(1207) and C32**

Register/Field		NFCM IO and Coherent Link	IOMMU IO Link	IOMMU Coherent Link
F3x140	FreeTok	8	8	8
	IsocRspTok	0	1	1
	IsocPreqTok	0	0	0
	IsocReqTok	0	1	1
	DnRspTok	1	1	1
	UpRspTok	3	3	3
	DnPreqTok	1	1	1
	UpPreqTok	1	1	1
	DnReqTok	1	1	1
	UpReqTok	2	2	2
F3x144	ProbeTok	3 <sup>1</sup> /1 <sup>2</sup>	3 <sup>1</sup> /1 <sup>2</sup>	3 <sup>1</sup> /1 <sup>2</sup>
	RspTok	3 <sup>1</sup> /5 <sup>2</sup>	3 <sup>1</sup> /5 <sup>2</sup>	3 <sup>1</sup> /5 <sup>2</sup>
F3x1[54, 50, 4C, 48]	IsocRspTok0	0	0	0
	IsocPreqTok0	0	0	0
	IsocReqTok0	0	1	1
	ProbeTok0	2	0	2
	RspTok0	2	2	2
	PReqTok0	2	2	2
	ReqTok0	2	2	2
	FreeTok	3	2	2
1. If probe filter is not enabled 2. If probe filter is enabled				

**Table 27: F3x140 and F3x144 Recommended Settings for Socket G34**

Register/Field	NFCM	IOMMU 2 Socket, Half Populated <sup>1</sup>	IOMMU 2 Socket, Fully populated <sup>1</sup>	IOMMU 4 Socket, Half populated <sup>2</sup>	IOMMU 4 Socket, Fully populated <sup>2</sup>	
F3x140	FreeTok	8 <sup>3</sup> /10 <sup>4</sup>	10 <sup>5</sup> /9 <sup>6</sup>	5 <sup>5</sup> /7 <sup>6</sup>	8	2 <sup>5</sup> /6 <sup>6</sup>
	IsocRspTok	0	1	1	1	1
	IsocPreqTok	0	0	0	0	0
	IsocReqTok	0	1	1	1	1
	DnRspTok	1	1	1	1	1
	UpRspTok	3	3	3 <sup>5</sup> /1 <sup>6</sup>	3	3 <sup>5</sup> /1 <sup>6</sup>
	DnPreqTok	1	1	1	1	1
	UpPreqTok	1	1	1	1	1
	DnReqTok	1	1	1	1	1
UpReqTok	1	1	1	1	1	
F3x144	ProbeTok	3 <sup>5</sup> /1 <sup>6</sup>	3 <sup>5</sup> /1 <sup>6</sup>	3 <sup>5</sup> /1 <sup>6</sup>	3 <sup>5</sup> /1 <sup>6</sup>	3 <sup>5</sup> /1 <sup>6</sup>
	RspTok	3 <sup>5</sup> /5 <sup>6</sup>	3 <sup>5</sup> /4 <sup>6</sup>	3 <sup>5</sup> /5 <sup>6</sup>	3 <sup>5</sup> /5 <sup>6</sup>	3 <sup>5</sup> /4 <sup>6</sup>

1. See Figure 5.
2. See Figure 6.
3. For a 2 processor system
4. For a 1 or 4 processor system
5. If probe filter is not enabled
6. If probe filter is enabled

**Table 28: F3x1[54, 50, 4C, 48] Recommended Settings for 1 Processor G34 System**

Register/Field	NFCM Ganged Link	NFCM Unganged Link <sup>3</sup>	IOMMU IO Link	IOMMU Ganged Coherent Link	IOMMU Unganged Coherent Link <sup>3</sup>	
F3x1[54, 50, 4C, 48]	IsocRspTok0	0	0	0	0	
	IsocPreqTok0	0	0	0	0	
	IsocReqTok0	0	0	1	1	
	ProbeTok0	2	1	0	2 <sup>1</sup> /1 <sup>2</sup>	1
	RspTok0	2	1	2	2	1
	PReqTok0	2	1	2	1 <sup>1</sup> /2 <sup>2</sup>	1
	ReqTok0	2	1	2	2	1
	FreeTok	0	0	0	0	0

1. If probe filter is not enabled
2. If probe filter is enabled
3. Unganged link settings should be used for both Tok0 and Tok1 fields

**Table 29: F3x1[54, 50, 4C, 48] Recommended Settings for 2 Processor G34 System**

Register/Field	NFCM Ganged Link	NFCM Unganged Link <sup>3</sup>	IOMMU IO Link	IOMMU Ganged Coherent Link	IOMMU Unganged Coherent Link <sup>3</sup>	
F3x1[54, 50, 4C, 48]	IsocRspTok0	0	0	0	0	
	IsocPreqTok0	0	0	0	0	
	IsocReqTok0	0	0	1	1	
	ProbeTok0	2	1	0	2 <sup>1</sup> /1 <sup>2</sup>	1
	RspTok0	2	1	2	2	1
	PReqTok0	2	1	2	1 <sup>1</sup> /2 <sup>2</sup>	1
	ReqTok0	2	1	2	2	1
	FreeTok	0	1	0	0	1
1. If probe filter is not enabled 2. If probe filter is enabled 3. Unganged link settings should be used for both Tok0 and Tok1 fields						

**Table 30: F3x1[54, 50, 4C, 48] Recommended Settings for 4 Processor G34 System**

Register/Field	NFCM Ganged Link	NFCM Unganged Link <sup>3</sup>	IOMMU IO Link	IOMMU Ganged Coherent Link	IOMMU Unganged Coherent Link <sup>3</sup>	
F3x1[54, 50, 4C, 48]	IsocRspTok0	0	0	0	0	
	IsocPreqTok0	0	0	0	0	
	IsocReqTok0	0	0	2	1	1
	ProbeTok0	2	1	0	2	1
	RspTok0	2	1	2	2 <sup>1</sup> /1 <sup>2</sup>	1
	PReqTok0	2	1	2	1	1
	ReqTok0	2	1	2	2	1
	FreeTok	0	0	0	0	0
1. If probe filter is not enabled 2. If probe filter is enabled 3. Unganged link settings should be used for Tok0 and Tok1 fields						

### 2.6.5 The Level 3 Cache (L3)

The NB may include an L3 cache as specified by [The L2/L3 Cache and L2 TLB Identifiers] CPUID Fn8000\_0006\_EDX.

When the L3 is enabled, the following register settings are required:

- [The Hardware Configuration Register (HWCR)] MSRC001\_0015[INVD\_WBINVD]=1.

#### 2.6.5.1 Probe Filter

The probe filter provides filtering of broadcast probes to improve link bandwidth and performance for multi-node systems. The probe filter is not supported for single link systems. It uses a portion of the L3 data cache as a directory to track cache lines in the system. The probe filter directory on each node tracks addresses mapped to the local DRAM on that node. The probe filter is enabled by programming [The Probe Filter Control Regis-



ter] F3x1D4. The probe filter is not supported if the processor does not contain an L3 cache. The probe filter should be enabled or disabled on all nodes in the system. The following register settings must be used in systems where the probe filter is enabled:

- MSRC001\_102A[ProbeFilterSupEn]=1.
- F2x1B0[CohPrefPrbLmt]=000b.

BIOS should enable the probe filter if either of the following conditions are true:

- F0x60[NodeCnt] equals 01h, both nodes have memory attached, and the MEMCLK frequencies are at least 533 MHz.
- F0x60[NodeCnt] equals 02h or greater.

Hardware initializes the probe filter directory when the probe filter is enabled. Initialization is complete when F3x1D4[PFInitDone] is set. The system must be quiesced of all coherent traffic during probe filter initialization. BIOS should perform the following steps to initialize the probe filter:

1. Disable the L3 and DRAM scrubbers on all nodes in the system:
  - F3x58[L3Scrub]=00h.
  - F3x58[DramScrub]=00h.
  - F3x5C[ScrubRedirEn]=0.
2. Wait 40us for outstanding scrub requests to complete.
3. Disable all cache activity in the system by setting CR0.CD for all active cores in the system.
4. Issue WBINVD on all active cores in the system.
5. Set F3x1C4[L3TagInit]=1.
6. Wait for F3x1C4[L3TagInit]=0.
7. Set F3x1D4[PFMode]=10b.
8. Wait for F3x1D4[PFInitDone]=1.
9. Enable all cache activity in the system by clearing CR0.CD for all active cores in the system.
10. Restore L3 and DRAM scrubber register values.

#### 2.6.5.1.1 Probe Filter Errors

The L3 data cache used to store probe filter directory data is ECC protected. Single-bit (correctable) errors are corrected and logged in [The MCA NB Status High Register] F3x4C. Multi-bit (uncorrectable) errors are logged in F3x4C, and the faulty probe filter location is no longer used. The directory is periodically scrubbed and the L3 scrubber may be redirected to scrub a location on correctable errors.

Uncorrectable ECC errors in the directory are not fatal, do not set F3x4C[UC], and do not cause machine check exceptions, but the directory entry is no longer reliable. The L3 reports uncorrectable ECC errors to the probe filter. Once such an error occurs, the faulty location index in the directory is no longer used and all requests which map to that index issue broadcast probes. F3x1D4[PFeccError] is set on an uncorrectable ECC error and can only be cleared by software. An interrupt may be generated on uncorrectable ECC errors by configuring F3x1D4[PFErrInt].

#### 2.6.6 Memory Scrubbers

The processor includes memory scrubbers specified in [The Scrub Rate Control Register] F3x58 and F3x5C. 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. The system memory scrubber is also employed as specified in [The On-Line Spare Control Register] F3xB0[SwapEn0].

Systems that enable scrubbing may wish to configure data cache and L2 scrubbing to operate, even when the core is halted (in the ACPI-defined C1 state). This is accomplished by programming [The ACPI Power State

**Control Registers] F3x[84:80][ClkDivisor]** associated with C1 to a divisor no deeper than divide-by-16; divisors of 16, 8, 4, 2, and 1 support scrubbing while the core is halted. If a deeper clock divisor is desired for C1 and the duration of halt states is relatively short lived, it is a minimum exposure for scrubbing to be suspended during the halt.

For recommendations on scrub rates, see 2.12.1.7 [Scrub Rate Considerations].

## 2.6.7 Physical Address Space

The processor supports 48 address bits of coherent memory space (256 terabytes) as indicated by [The Address Size And Physical Core Count Information] CPUID Fn8000\_0008\_EAX. The processor master aborts the following upper-address transactions (to address PhysAddr):

- IO link requests with non-zero PhysAddr[63:48].
- IO link or CPU requests with non-zero PhysAddr[47:40] where F0x68[CHtExtAddrEn]=0.
- IO link or CPU requests with non-zero PhysAddr[47:40] which targets an IO link for which the appropriate F0x[E4, C4, A4, 84][Addr64BitEn]=0.
- IO link requests with non-zero PhysAddr[47:40] received from an IO link for which the appropriate F0x[E4, C4, A4, 84][Addr64BitEn]=0.

## 2.6.8 System Address Map

System software must not map memory in the reserved HyperTransport™ technology address regions. The *HyperTransport™ I/O Link Specification* details the address map available to system hosts and devices. Downstream host accesses to reserved HyperTransport™ address regions result in a page fault. Upstream system device accesses to reserved HyperTransport™ address regions result in undefined operation.

## 2.7 Links

A *link* is a block of link signals, including 16 CAD signals, 2 CTL signals, and 2 CLK signals. Links may support *unganged* modes in which subgroups of link signals--or *sublinks*--are connected to separate devices, as specified by [The Northbridge Capabilities Register] F3xE8[UnGangEn]. Links may operate per coherent protocol or IO protocol. The electrical definition is per various revisions of the *HyperTransport™ I/O Link Specification*; the terminology for these modes is as follows:

- Gen1: refers to link rates of 0.4 to 1.6 GT/s in the revision 1 specification or 2.0 GT/s in the revision 2 specification.
- Gen3: refers to link rates of 2.4 to 6.4 GT/s in the revision 3 specification.

### 2.7.1 Link Initialization

#### 2.7.1.1 Ganging And Unganging

The following combinations of maximum bit widths (it is always possible to connect to a device using a supported, narrower bit width), protocols, and frequencies are supported:

- One 16-bit link (ganged); either IO or coherent protocol; any supported link frequency. In ganged mode, the link may or may not be left unconnected. In ganged mode, registers that control sublink 0 control the entire link; registers that control sublink 1 are reserved.
- Two 8-bit links (unganged); the two sublinks may be configured for any combination of IO or coherent protocol; if the two link frequencies are the same, then any supported frequency is allowed; if the two link frequencies are different, then the frequencies are required to be one of the following ratios: 8:1, 6:1, 4:1, 2:1; legal combinations are {5.6, 2.8}, {6.4, 3.2}, {6.4, 1.6}, {4.8, 2.4}, {4.8, 1.2}, {4.0, 2.0}, {3.2, 1.6}, {2.4,

1.2}, and {0.8, 0.4} GT/s). The following combinations are allowed for test and debug purposes: {6.4, 0.8}, {2.4, 0.4}, {4.8, 0.8}, {3.2, 0.8}, {3.2, 0.4}, {1.6, 0.8}, and {1.6, 0.4}. In ungang mode, neither, either, or both of the two 8-bit sublinks may be left unconnected. In ungang mode, sublink 0 refers to the link associated with CLK[0], CTL[0], and CAD[7:0]; sublink 1 refers to the link associated with CLK[1], CTL[1], and CAD[15:8].

### 2.7.1.2 Ganging Detection And Control

If unganging is not supported by the processor (F3xE8[UnGangEn]), then the links always cold boot to the ganged state. See Table 2.

Otherwise, the ganged state at cold boot is based on the state of CTL[1]. If CTL[1]=0, then the link powers up ungang. If CTL[1]=1, then the link powers up ganged. If CTL[1] is connected between the processor and another device (such as another processor) that supports the Gen3 link specification, then the link cold boots to the ungang state.

If both sublinks of an ungang link connect the same two devices, then initialization software may be used to place these sublinks into the ganged state (F0x[18C:170][Ganged]).

Unused upper sublinks should be left disconnected with processor inputs floating to minimize power consumption. Software should not gang inactive sublinks with active sublinks.

### 2.7.1.3 Link Type Detect

The link may be initialized in one of the following states during cold reset:

- The link may be ganged or ungang.
- The link/sublink is not connected with inputs terminated to the proper state to indicate ganged/ungang.
- The link/sublink is not connected with inputs floating (as with a connection to an unpopulated socket).

The processor follows the protocol described in the Gen3 link specification to determine the cold boot state of F0x[18C:170][Ganged] and F0x[E4, C4, A4, 84][TransOff and EndOfChain]. F0x[E4, C4, A4, 84][TransOff and EndOfChain] are set when the link is unconnected, as follows:

- Un-terminated link: no device is detected on the other side of the link.
- DC-coupled links: link is strapped in the unconnected state per the link specification.

#### 2.7.1.3.1 Unused Links

For revision C and later revisions, when both sublinks of a link are unused and disconnected, the clocks to the link are disabled and the following registers corresponding to the unused link become unavailable. Writes to these registers are ignored and reads are undefined.

- F3x1[54, 50, 4C, 48].
- F0x[F0, D0, B0, 90] and F4x[F0, D0, B0, 90].
- F0x[F4, D4, B4, 94] and F4x[F4, D4, B4, 94].
- F4x1[98, 90, 88, 80].
- F4x1[9C, 94, 8C, 84].

### 2.7.1.4 Legal Topologies

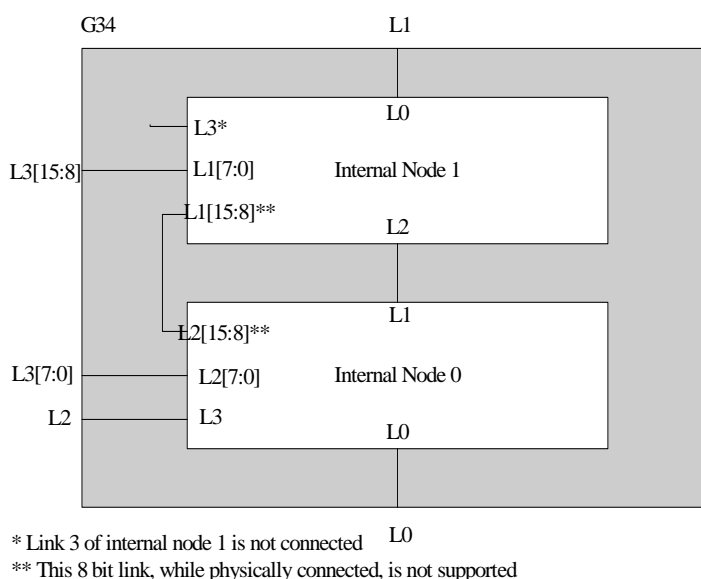
The link may be connected in these configurations:

- 16-bit Gen3 device connected (CTL[1] connected)
- 16-bit Gen1 device connected (CTL[1] terminated)

- Unganged:
  - Two 8-bit devices connected
  - One 8-bit device connected to either sublink and the other sublink with inputs terminated
  - One 8-bit device connected to either sublink and the other sublink with inputs floating
- Link inputs terminated
- Link inputs floating

### 2.7.1.5 Link Mapping for Dual-Node Processors in the G34 Package

The G34 package supports dual-node processors (`F3xE8[MultiNodeCpu]=1`). The links on the G34 package are connected to the links of the internal nodes as shown in Figure 9. See 2.6.4.2 [HyperTransport™ Technology Routing] for supported system topologies.



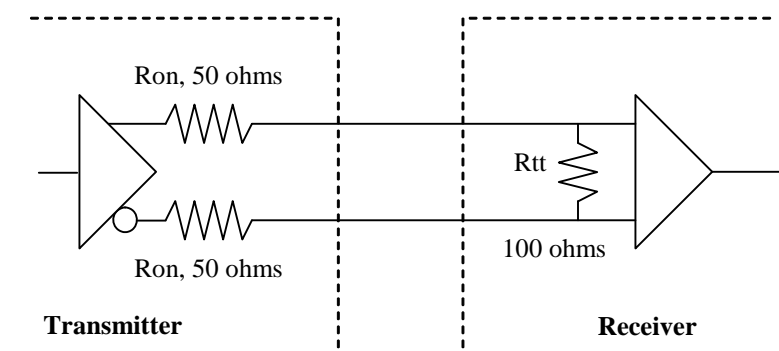
**Figure 9: Dual-node processor link diagram.**

Since the internal 8 bit link is not supported, BIOS must do the following for dual-node processors:

- Program `F0x16C[ConnDly]=1` for both internal nodes.
- If `F3xE8[IntNodeNum]=00b`, program `F4xC4[TransOff and EndOfChain]=1`. See `F4x[E4, C4, A4, 84]`.
- If `F3xE8[IntNodeNum]=01b`, program `F4xA4[TransOff and EndOfChain]=1`. See `F4x[E4, C4, A4, 84]`.

### 2.7.2 Termination and Compensation

The links are designed to operate in DC-termination mode as follows.



**Figure 10: Link DC termination mode.**

$R_{on}$  and  $R_{tt}$  are constructed with an array of parallel resistors that can be enabled or disabled to vary the resulting resistance. Each parallel resistor is referred to as a *tap*. Precision external resistors are used by the processor to determine the number of taps that must be enabled in order to match  $R_{on}$  and  $R_{tt}$  to the proper target values. The results of this compensation circuitry are observable in [The Link Phy Compensation Control Register] F4x1[9C, 94, 8C, 84]\_xE0[RonRawCal] and [The Link Phy Compensation Control Register] F4x1[9C, 94, 8C, 84]\_xE0[RttRawCal]. Other fields in these registers are provided to offset the raw calculated compensation values or override them.

Compensation updates start after PWROK becomes valid (and occur while RESET\_L is asserted).

The transmitter and receiver tristate in the PHY OFF state, as entered by F0x[E4, C4, A4, 84][TransOff]=1.

### 2.7.3 Equalization

A high speed data stream passing through the channel distorts due to various effects. The processor employs equalization to counter this problem and to improve electrical fidelity of the links. Equalization is employed by changing the voltage level transmitted before and after bit transitions. The transmitter can be attenuated to levels that vary based on bit history, as specified by [The Link Phy Deemphasis Value Registers] F4x1[9C, 94, 8C, 84]\_x[D5, C5]. Equalization is not used at Gen1 frequencies.

### 2.7.4 Link Bandwidth Requirements

The bandwidth of a link may not exceed the bandwidth capacity of the node's NB. The requirements are as follows:

- For revision C and earlier revisions and revision E where NCLK is the NB COF and HTCLK is the frequency of the link clock:
  - 16-bit, ganged links or any ganged links running link BIST:  $NCLK \geq HTCLK$ .
  - 8-bit or less, ganged links:  $NCLK \geq HTCLK / 2$ .
  - All unganged links:  $NCLK \geq HTCLK$ .
- For revision D, where NCLK is the NB COF and HTCLK is the frequency of the link clock:
  - F0x[14C:130][RetryModeEnable]=1:  $NCLK \geq HTCLK / 2$ .
  - F0x[14C:130][RetryModeEnable]=0:  $NCLK \geq HTCLK$ .

### 2.7.5 Link Retry

The links support the error-retry mode described by the *HyperTransport™ I/O Link Specification*, controlled by [The Link Retry Registers] F0x[14C:130] and [The Link Global Retry Control Register] F0x150. Some requirements for operation in this mode:

- The processor does not support error-retry mode over links operating at Gen1 frequencies.
- IO links operating at Gen3 frequencies are required to have error-retry enabled.
- Coherent links operating at Gen3 frequencies are expected to have error-retry enabled. However, the processor logically supports operation of coherent links at Gen3 frequencies without error-retry for test and debug purposes.
- If any coherent links have error-retry enabled, then all coherent links are required to have it enabled after the coherent fabric has been configured. Until the warm reset to enable retry on all links, only configuration space cycles may be used in the coherent fabric.
- The retry history buffer for each ganged link supports up to 32 packets (each packet may include command and data), 16 packets for each unganged sublink.

### 2.7.6 Link LDTSTOP\_L Disconnect-Reconnect

When disconnected for an LDTSTOP\_L assertion, the state of the link and the reconnect time is a function of the link generation (Gen1 or Gen3) being used (or that the link is changing to, as a result of the LDTSTOP\_L assertion), `F0x[E4, C4, A4, 84][LdtStopTriEn]`, and `F0x[18C:170][LS2En]` as follows:

**Table 31: Link disconnect controls**

Link Gen	LdtStopTriEn	LS2En	CLK	CAD, CTL	Reconnect delay
Gen1	0	X	L0 <sup>1</sup>	L0 <sup>1</sup>	Fast (about 1 microsecond) <sup>4</sup>
Gen1	1	0	L0 <sup>1</sup>	High imp <sup>3</sup>	Fast (about 1 microsecond) <sup>4</sup>
Gen1	1	1	High imp <sup>3</sup>	High imp <sup>3</sup>	<code>F3xD8[ReConDel]</code> <sup>4</sup>
Gen3	X	0	L0 <sup>1</sup>	EI <sup>2</sup>	<code>F0x16C[T0Time]</code>
Gen3	X	1	EI <sup>2</sup>	EI <sup>2</sup>	<code>F0x16C[T0Time]</code>

1. L0 represents the active, driven state.
2. Electrical idle.
3. High impedance.
4. `F0x[E4, C4, A4, 84][ExtCTL]=1` adds 50us after CTL asserts.

### 2.7.7 LDTSTOP Requirements

- The processor requires additional minimum LDTSTOP\_L assertion time for certain system configurations.
  - If any of the following system configuration properties are true the minimum LDTSTOP\_L assertion time required by the processor is 10 microseconds:
    - The system includes links operating at 200 MHz, 400 MHz, or 600 MHz (see [\[The Link Frequency/Revision Registers\] F0x\[E8, C8, A8, 88\]\[Freq\]](#)).
    - The system connects to registered DIMMs (see [\[The DRAM Configuration Low Register\] F2x\[1, 0\]90\[UnbuffDimm\]](#)).
    - The system supports the C1E state, `F3x[84:80][CpuPrbEn]` for C1E is 0, and the link is operating at a Gen1 frequency.
    - Revision D and later processors.
  - For all other configurations the minimum LDTSTOP\_L assertion time is as specified by the link specification (1 microsecond).
- For all cases of LDTSTOP\_L assertion LDTSTOP\_L must not deassert less than 10 microseconds after the processor broadcasts the STOPGRANT message.
  - For revision D and later processors with an NB COF < 1.4 GHz, LDTSTOP\_L must not deassert less than 16 microseconds after the processor broadcasts the STOPGRANT message.
- The processor requires a minimum LDTSTOP\_L deassertion time as follows:
  - 5 us for unbuffered DIMMs and `F2x[1, 0]90[DisDIIShutdownSR]=1`.

- 10 us for unbuffered DIMMs and  $F2x[1, 0]90[DisDIIShutdownSR]=0$ .
  - 3 us plus  $F2x[1, 0]90[PIILockTime]$  for registered DDR2 DIMMs.
  - 11 us for registered DDR3-1333 DIMMs.
  - 12 us for registered DDR3-1066 DIMMs.
  - 15 us for registered DDR3-800 DIMMs.
  - 18 us for registered DDR3-667 DIMMs.
- Narrow or slow links and the use of  $F0x[E4, C4, A4, 84][ExtCTL]$  can greatly increase the time for a Gen1 link to disconnect and reconnect, so the time between LDTSTOP assertions must be increased appropriately as required by section 8.3 of the *HyperTransport™ I/O Link Specification*.

### 2.7.8 Response Ordering

The processor supports non-standard response ordering, not required by the link specification. If the processor receives multiple IO-sourced memory read requests with certain attributes, then the processor ensures that the order of the responses to these requests is the same as the order in which the requests were received. The required attributes are:

- The requests have the same UnitID value (or *logical* UnitID if multiple UnitIDs are clumped; see [\[The Link Clumping Enable Registers\] F0x\[11C, 118, 114, 110\]](#)).
- The requests have the same, non-zero SeqID value.
- The requests have the same PassPW bit value.
- The requests have the same Coherent (snoop) bit value.
- The requests have the same RespPassPW bit value.
- The requests have the same Normal/Isochronous bit value.

This feature may allow IO devices to be designed that do not require re-order buffers. This behavior may be disabled through [\[The Northbridge Configuration Register \(NB\\_CFG\)\] MSRC001\\_001F\[DisOrderRdRsp\]](#).

### 2.7.9 Link Testing, BIST, and ILM

The processor includes a link-defined BIST engine for each link. The control registers are found starting at [\[The Link BIST Control Register\] F4x1\[9C, 94, 8C, 84\]\\_x100](#). See the link specification for more information.

The processor also supports link-defined internal loopback mode (ILM), controlled by [\[The Link Extended Control Registers\] F0x\[18C:170\]\[ILMEn\]](#).

### 2.7.10 Miscellaneous Behaviors and Requirements

- The processor does not support the link-defined Atomic read-modify-write command and returns target abort for any that are received.
- The processor does not support Device Messages and returns master abort for any that are received.
- The processor ignores the Chain bit.
- The processor checks for differential signaling on CTL[1:0] and disabled unused sublinks.
- Revision C processors that support [MSRC001\\_001F\[GfxNbPstateDis\]](#) and have NB P-states enabled require the chipset to manage the link width via LDTSTOP to meet link bandwidth requirements. See [2.7.4 \[Link Bandwidth Requirements\]](#). For all other processors link width and frequency changes via LDTSTOP assertion are only supported by BIOS when the system is quiesced.
- Revision D0 processors support LDTSTOP assertion for S1 and S3 only.
- The processor register space does not include the Gen3 link-defined UCC bit or CPIC bit. However, functionally, the initial revisions of the processor would have these bits set to indicate that unthrottled command generation from IO links is supported (i.e., setting [LinkTrain\[DisCmdThrt\]](#) on the other side of the link) and



command packet insertion from IO links is supported (i.e., setting LinkTrain[CPIEn] on the other side of the link). However, no assurances are made regarding future processor revisions; they may rely on throttling and disabled command packet insertion to operate.

- While transmitting to an IO link, the processor does not ever insert commands (other than NOPs) into data packets and the processor supports throttling command generation based on the state of F0x168[DisNcHtCmdThrottle].
- The processor logically supports link-defined mode combinations as follows (however electrical requirements may limit some options):

**Table 32: Supported link operational modes**

Frequency	200-1000MHz	1200MHz and higher		
<b>Coupling/ Link Type</b>	DC	DC non-coherent operational	DC coherent operational	DC test/debug
<b>Termination</b>	RXDIF	RXDIF	RXDIF	RXDIF
<b>8b10b</b>	No	No	No	No
<b>Scrambling</b>	No	Required	Required	Optional
<b>Gen3 Training</b>	No	Yes	Yes	Yes
<b>Retry</b>	No	Required	Optional at low speed	Optional

- The processor supports link-defined INTx messages. It emulates the ORing of INTx assertions throughout the system and broadcasts the result. To accomplish this, the processor uses separate counters for each of the four interrupts (INTA, INTB, INTC, and INTD) which track INTx assertions and deassertions received by the coherent fabric. Each assertion causes the counter to increment and each deassertion causes the counter to decrement. As each counter transitions from 0 to 1, the interrupt assertion message is broadcast. As each counter transitions from 1 to 0, the interrupt deassertion message is broadcast.
- The processor reflects system management messages E2h to FFh for vendor-defined virtual wire messages. Devices that send or receive them must have programmable registers to control the command encodings used so that different devices can interoperate.
- Ganged links leave the upper sublink driven after cold reset (per F0x16C[InLnSt]) unless the lower sublink is unconnected. If the lower sublink of a ganged link is unconnected, the entire link is disabled.
- The processor cannot be used in a system where the sideband signal (RESET# or LDTSTOP#) skew between devices is greater than 100us.
- The processor only supports synchronous clocking mode, where both sides of the link have their clocks derived from the same oscillator.
- No ordering may be assumed between broadcast requests and posted writes.
- The processor allows reordering of upstream posted requests with PassPW=1.

## 2.8 DRAM Controllers (DCTs)

The DCTs support DDR2 DIMMs or DDR3 DIMMs. Products may be configurable between DDR2 and DDR3 operation. See [Table 2](#).

A *DRAM channel* is the group of the DRAM interface pins that connect to one series of DIMMs. The processor supports two DDR channels in single-node products and four DDR channels in dual-node products. The processor includes two DCTs per node. Each DCT controls one 64-bit DDR DIMM channel.

For single-node products, DCT0 controls channel A DDR pins and DCT1 controls channel B DDR pins. However, the processor may be configured: (1) to behave as a single dual-channel DCT; this is called *ganged mode*;

or (2) to behave as two single-channel DCTs; this is called *unganged mode*.

For dual-node products, DCT0 of internal node 0 controls channel A DDR pins, DCT1 of internal node 0 controls channel B DDR pins, DCT0 of internal node 1 controls channel C DDR pins, and DCT1 of internal node 1 controls channel D DDR pins. Ganged mode may only be configured for DCTs on the same node.

BIOS reads [F3xE8\[MultiNodeCpu\]](#) and [F3xE8\[IntNodeNum\]](#) to determine the DCT to DDR channel mapping.

A *logical DIMM* is either one 64-bit DIMM (as in unganged mode) or two identical DIMMs in parallel to create a 128-bit interface (as in ganged mode). See [2.8.5 \[Ganged or Unganged Mode Considerations\]](#).

There are restrictions on the configuration and types of DIMMs supported on the DCTs at any one time:

- All DIMMs connected to a node are required to operate at the same MEMCLK frequency, regardless of which channel they are connected to. Both DCTs must be programmed to the same frequency.
- Mixing of DRAM types (DDR2 and DDR3) within a system is not supported.
- Mixing of unbuffered and registered DIMMs within a system is not supported.
- Mixing of ECC and non-ECC DIMMs within a system is not supported.

[Table 33](#) and [Table 35](#) below list the maximum DIMM speeds supported by the processor for different configurations. See [2.8.9.5.8 \[DRAM Address Timing and Output Driver Compensation Control\]](#) for detailed information on supported memory bus loads and for configuration settings based on loads.

**Table 33: DDR2 Unbuffered and Registered Maximum Frequency Support (per channel)**

DIMM Slots	DIMMs	DIMMs		Frequency (MT/s)		
		QR	SR or DR	SO-DIMM	Unbuffered DIMM	Registered DIMM
1	1	0	1	800	1066	800
		1	0	-	-	533
2	1	0	1	800	1066	800
		1	0	-	-	533
	2	0	2	800	800	800
		2	0	-	-	533
4	1 or 2	-	any	-	-	800
4	3 or 4	-	any	-	-	533

**Table 34: DDR3 SO-DIMM and Unbuffered Maximum Frequency Support (per channel)**

DIMM Slots	DIMMs	Frequency (MT/s)	
		SO-DIMMs on motherboard	Unbuffered DIMMs on motherboard
1	1	1333	1333
2 or 3	1 or 2	1066	1333

Notes:

- 2 and 3 DIMM slot boards require a population order restriction for partially populated channels (see 2.8.9.5.8 [DRAM Address Timing and Output Driver Compensation Control]).
- 3 DIMM slots are supported only on processors in the G34 package. A maximum of 2 unbuffered DIMMs populated per channel is permitted.

**Table 35: DDR3 Registered Maximum Frequency Support (per channel)**

DIMM Slots <sup>1</sup>	DIMMs	DIMMs		Frequency <sup>2</sup> (MT/s)	
		QR	SR or DR	1.5V	1.35V
1	1	-	1	1333	1333
		1	-	1333	1333
2	1	-	1	1333	1333
		1	-	1333	1066
	2	-	2	1333	1333
		1 or 2	1 or 0	1066	800
3	1	-	1	1333	1333
		1	-	1066	1066
	2	-	2	1333	1333
		1	1	800	800
	3	-	3 SR	1333	1066
		-	3 DR	1066	800
		-	3 mixed	1066	800
		1	2	800	667

1. 3 DIMM slots are supported only on processors in the G34 and C32 packages.  
2. Frequency, voltage, and population restrictions (including the order for partially populated channels) may apply (see 2.8.9.5.8 [DRAM Address Timing and Output Driver Compensation Control]).

### 2.8.1 DCT Configuration Registers

DCT configuration registers range from F2x[1, 0][5C:40] through F2x[1, 0]A8 and F2x110 through F2x11C. F2x0XX registers are associated with DCT0 and F2x1XX registers are associated with DCT1.

If the BIOS is not required to be in compatibility mode with the NPT Family 0Fh memory controller, BIOS must program F2x[1, 0]94[LegacyBiosMode]=0.

When the DCTs are ganged, as specified by [The DRAM Controller Select Low Register] F2x110[DctGangEn], then most of the DCT configuration registers behave as follows: only writes the DCT0 set of registers

(F2x0XX) are captured (and applied to both channels); reads to the DCT0 set of registers return the value captured in the DCT0 channel registers; writes to the DCT1 set of registers (F2x1XX) are ignored and reads return all 0's. The exception is the DCT phy registers, F2x[1, 0]98, F2x[1, 0]9C, and all the associated indexed registers; these all remain independently accessible between the two DCTs when the DCTs are ganged.

## 2.8.2 DDR Pad to Processor Pin Mapping

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

**Table 36: Package pin mapping**

Pad	Pin <sup>1</sup>					
	F(1207)	AM2r2	AM3/ASB2 S1g3/S1g4	C32	G34 Internal Node 0	G34 Internal Node 1
MEMCLK0_H[0]	-	MA1_CLK_H[1]	MA_CLK_H[0] <sup>2</sup>	-	-	MC_CLK_H[4]
MEMCLK0_H[1]	-	MA0_CLK_H[1]	MA_CLK_H[1]	-	-	-
MEMCLK0_H[2]	MA3_CLK	-	MA_CLK_H[2] <sup>2</sup>	MA_CLK_H[3]	MA_CLK_H[0]	MC_CLK_H[0]
MEMCLK0_H[3]	MA2_CLK	-	MA_CLK_H[3] <sup>2</sup>	MA_CLK_H[2]	MA_CLK_H[1]	MC_CLK_H[1]
MEMCLK0_H[4]	MA1_CLK	MA1_CLK_H[0]	MA_CLK_H[4]	MA_CLK_H[1]	MA_CLK_H[2]	MC_CLK_H[2]
MEMCLK0_H[5]	MA0_CLK	MA0_CLK_H[0]	MA_CLK_H[5]	MA_CLK_H[0]	MA_CLK_H[3]	MC_CLK_H[3]
MEMCLK0_H[6]	-	MA1_CLK_H[2]	MA_CLK_H[6] <sup>2</sup>	-	MA_CLK_H[4]	-
MEMCLK0_H[7]	-	MA0_CLK_H[2]	MA_CLK_H[7]	-	-	-
MEMCLK1_H[0]	-	MB1_CLK_H[1]	MB_CLK_H[0] <sup>2</sup>	-	-	MD_CLK_H[4]
MEMCLK1_H[1]	-	MB0_CLK_H[1]	MB_CLK_H[1]	-	-	-
MEMCLK1_H[2]	MB3_CLK	-	MB_CLK_H[2] <sup>2</sup>	MB_CLK_H[3]	MB_CLK_H[0]	MD_CLK_H[0]
MEMCLK1_H[3]	MB2_CLK	-	MB_CLK_H[3] <sup>2</sup>	MB_CLK_H[2]	MB_CLK_H[1]	MD_CLK_H[1]
MEMCLK1_H[4]	MB1_CLK	MB1_CLK_H[0]	MB_CLK_H[4]	MB_CLK_H[1]	MB_CLK_H[2]	MD_CLK_H[2]
MEMCLK1_H[5]	MB0_CLK	MB0_CLK_H[0]	MB_CLK_H[5]	MB_CLK_H[0]	MB_CLK_H[3]	MD_CLK_H[3]
MEMCLK1_H[6]	-	MB1_CLK_H[2]	MB_CLK_H[6] <sup>2</sup>	-	MB_CLK_H[4]	-
MEMCLK1_H[7]	-	MB0_CLK_H[2]	MB_CLK_H[7]	-	-	-
MEMCS0_L[0]	MA0_CS_L[0]					MC0_CS_L[0]
MEMCS0_L[1]	MA0_CS_L[1]					MC0_CS_L[1]
MEMCS0_L[2]	MA1_CS_L[0]					MC1_CS_L[0]
MEMCS0_L[3]	MA1_CS_L[1]					MC1_CS_L[1]
MEMCS0_L[4]	MA2_CS_L[0]	-	-	MA0_CS_L[2]	MA2_CS_L[0]	MC2_CS_L[0]
MEMCS0_L[5]	MA2_CS_L[1]	-	-	MA0_CS_L[3]	MA2_CS_L[1]	MC2_CS_L[1]
MEMCS0_L[6]	MA3_CS_L[0]	-	-	MA1_CS_L[2]	MA3_CS_L[0]	MC3_CS_L[0]
MEMCS0_L[7]	MA3_CS_L[1]	-	-	MA1_CS_L[3]	MA3_CS_L[1]	MC3_CS_L[1]
MEMCS1_L[0]	MB0_CS_L[0]					MD0_CS_L[0]
MEMCS1_L[1]	MB0_CS_L[1]					MD0_CS_L[1]
MEMCS1_L[2]	MB1_CS_L[0]					MD1_CS_L[0]
MEMCS1_L[3]	MB1_CS_L[1]	-	MB1_CS_L[1] <sup>2</sup>	MB1_CS_L[1]		MD1_CS_L[1]
MEMCS1_L[4]	MB2_CS_L[0]	-	-	MB0_CS_L[2]	MB2_CS_L[0]	MD2_CS_L[0]

**Table 36: Package pin mapping**

Pad	Pin <sup>1</sup>					
	F(1207)	AM2r2	AM3/ASB2 S1g3/S1g4	C32	G34 Internal Node 0	G34 Internal Node 1
MEMCS1_L[5]	MB2_CS_L[1]	-	-	MB0_CS_L[3]	MB2_CS_L[1]	MD2_CS_L[1]
MEMCS1_L[6]	MB3_CS_L[0]	-	-	MB1_CS_L[2]	MB3_CS_L[0]	MD3_CS_L[0]
MEMCS1_L[7]	MB3_CS_L[1]	-	-	MB1_CS_L[3]	MB3_CS_L[1]	MD3_CS_L[1]
MEMODT0[0]	MA0_ODT[0]					MC0_ODT[0]
MEMODT0[1]	MA1_ODT[0]					MC1_ODT[0]
MEMODT0[2]	MA2_ODT[0]	-	MA0_ODT[1]	MA0_ODT[1]	MA2_ODT[0]	MC2_ODT[0]
MEMODT0[3]	MA3_ODT[0]	-	MA1_ODT[1]	MA1_ODT[1]	MA3_ODT[0]	MC3_ODT[0]
MEMODT1[0]	MB0_ODT[0]					MD0_ODT[0]
MEMODT1[1]	MB1_ODT[0]					MD1_ODT[0]
MEMODT1[2]	MB2_ODT[0]	-	MB0_ODT[1]	MB0_ODT[1]	MB2_ODT[0]	MD2_ODT[0]
MEMODT1[3]	MB3_ODT[0]	-	MB1_ODT[1] <sup>2</sup>	MB1_ODT[1]	MB3_ODT[0]	MD3_ODT[0]
MEMCKE0[0]	MA_CKE[0]					MC_CKE[0]
MEMCKE0[1]	MA_CKE[1]					MC_CKE[1]
MEMCKE1[0]	MB_CKE[0]					MD_CKE[0]
MEMCKE1[1]	MB_CKE[1]					MD_CKE[1]
1. For differential pins, only positive polarity pins are shown; negative polarity pins have corresponding mapping and are controlled by the same pad.						
2. AM3 and ASB2 only.						

### 2.8.3 Support For Multiple Unbuffered Logical DIMMs

There is one copy of command and address pins for each DRAM channel supported by the package. It is expected that the electrical requirements for unbuffered DIMMs necessitate that slow access mode ([[The DRAM Configuration High Register](#)] F2x[1, 0]94[SlowAccessMode]) be enabled when there is more than one unbuffered logical DIMM installed to a DRAM controller.

### 2.8.4 Burst Length

Some IO applications such as graphics may access system memory with many 32-byte transactions. In these cases, placing the DRAM controller into 32-byte burst mode ([[The DRAM Configuration Low Register](#)] F2x[1, 0]90[BurstLength32]) may improve DRAM efficiency. When a DRAM controller is programmed for 128-bit logical DIMMs (F2x[1, 0]90[Width128]) then only 64-byte bursts are supported.

### 2.8.5 Ganged or Unganged Mode Considerations

When the DCTs are in ganged mode, as specified by [[The DRAM Controller Select Low Register](#)] F2x110[DctGangEn], then each logical DIMM is two channels wide. Each physical DIMM of a 2-channel logical DIMM is required to be the same size and use the same timing parameters. Both DCTs must be programmed with the same information (see 2.8.1 [[DCT Configuration Registers](#)]). When the DCTs are in

unganged mode, a logical DIMM is equivalent to a 64-bit physical DIMM and each channel is controlled by a different DCT.

Typical systems are recommended to run in unganged mode to benefit from the additional parallelism generated by using the DCTs independently. See 2.12.2 [DRAM Considerations for ECC] for DRAM ECC implications of ganged and unganged mode. Ganged mode is not supported for S1g3, S1g4, ASB2, and G34 processors.

When enabling two DCTs in unganged mode, BIOS should set `F2x[1, 0]94[BankSwizzleMode]=1b`, `F2x110[DctSelIntLvAddr]=10b` for DDR2 or `F2x110[DctSelIntLvAddr]=11b` for DDR3, and `F2x110[DctSelIntLvEn]=1b`.

## 2.8.6 Routing DRAM Requests

Typically, system BIOS acquires DIMM configuration information, such as the amount of memory on each DIMM, from the Serial Presence Detect (SPD) ROM on each DIMM and uses this information to program the DRAM controller registers.

DRAM requests are mapped to the DCT of the appropriate node based on the routing configuration specified in section 2.6.4.1.1 [DRAM and MMIO Memory Space]. They are mapped to chip selects through [The DRAM CS Base Address Registers] `F2x[1, 0][5C:40]`, and [The DRAM CS Mask Registers] `F2x[1, 0][6C:60]`.

The following algorithm is designed to be used to determine the processor, the DRAM controller, and the chip select for a system address that maps to DRAM. `SystemAddr` is a 64 bit input variable representing the physical address. `CSFound`, `NodeID`, `ChannelSelect`, and `CS` are output variables. If `CSFound` is equal to 1, then `NodeID`, `ChannelSelect`, and `CS` outputs are equal to the node, DRAM controller (zero or one), and the chip select that corresponds to the input address.

If the On-line Spare feature is enabled BIOS assigns one of the chip-selects for a controller, `CH0SPARE_RANK` or `CH1SPARE_RANK`, to be the spare rank in the event of a DIMM failure precondition. If the DIMM failure precondition occurs and the data of the failing rank is copied over, the spare rank decodes to the same system address range as the failing rank (`BadDramCs`).

```
(int,int,int,int) TranslateSysAddrToCS((uint64)SystemAddr){
int SwapDone, BadDramCs;
int CSFound, NodeID, CS, F1Offset, F2Offset, F2MaskOffset, Ilog, device;
int HiRangeSelected, DramRange;
uint32 IntlvEn, IntlvSel;
uint32 DramBaseLow, DramLimitLow, DramEn;
uint32 HoleOffset, HoleEn;
uint32 CSBase, CSLimit, CSMask, CSEn;
uint32 InputAddr, Temp;
uint32 OnlineSpareCTL;
uint32 DctSelBaseAddr, DctSelIntLvAddr, DctGangEn, DctSelIntLvEn;
uint32 DctSelHiRngEn, DctSelHi;
uint64 DramBaseLong, DramLimitLong;
uint64 DctSelBaseOffsetLong, ChannelOffsetLong, ChannelAddrLong;

// device is a user supplied value for the PCI device ID of the processor
// from which CSRs are initially read from (current processor is fastest).
// CH0SPARE_RANK and CH1SPARE_RANK are user supplied values, determined
// by BIOS during DIMM sizing.
```

```

CSFound = 0;
for(DramRange = 0; DramRange < 8; DramRange++)
{
    F1Offset = 0x40 + (DramRange << 3);
    DramBaseLow = Get_PCI(bus0, device, func1, F1Offset);
    DramEn = DramBaseLow & 0x00000003;
    IntlvEn = (DramBaseLow & 0x00000700) >> 8;
    DramBaseLow = DramBaseLow & 0xFFFF0000;
    DramBaseLong = (((Get_PCI(bus0, device, func1, F1Offset + 0x100) & 0xFF)<<32) +
        DramBaseLow)<<8;
    DramLimitLow = Get_PCI(bus0, device, func1, F1Offset + 4);
    NodeID = DramLimitLow & 0x00000007;
    IntlvSel = (DramLimitLow & 0x00000700) >> 8;
    DramLimitLow = DramLimitLow | 0x0000FFFF;
    DramLimitLong = (((Get_PCI(bus0, device, func1, F1Offset + 0x104) & 0xFF)<<32) +
        DramLimitLow)<<8 | 0xFF;
    HoleEn = Get_PCI(bus0, dev24 + NodeID, func1, 0xF0);
    HoleOffset = (HoleEn & 0x0000FF80);
    HoleEn = (HoleEn & 0x00000003);
    if(DramEn && DramBaseLong <= SystemAddr && SystemAddr <= DramLimitLong)
    {
        if(IntlvEn == 0 || IntlvSel == ((SystemAddr >> 12) & IntlvEn))
        {
            if(IntlvEn == 1) Ilog = 1;
            else if(IntlvEn == 3) Ilog = 2;
            else if(IntlvEn == 7) Ilog = 3;
            else Ilog = 0;

            Temp = Get_PCI(bus0, dev24 + NodeID, func2, 0x10C);
            IntLvRgnSwapEn = temp & 0x1;
            if(IntLvRgnSwapEn)
            {
                IntLvRgnBaseAddr = Temp >> 3 & 0x7F;
                IntLvRgnLmtAddr = temp >> 11 & 0x7F;
                IntLvRgnSize = temp >> 20 & 0x7F;
                if((SystemAddr >> 34 == 0) &&
                    ((SystemAddr >> 27 >= IntLvRgnBaseAddr) &&
                    (SystemAddr >> 27 <= IntLvRgnLmtAddr) ||
                    (SystemAddr >> 27 < IntLvRgnSize)))
                {
                    SystemAddr ^= IntLvRgnBaseAddr << 27;
                }
            }

            Temp = Get_PCI(bus0, dev24 + NodeID, func2, 0x110);
            DctSelHiRngEn = Temp & 1;
            DctSelHi = Temp>>1 & 1;
            DctSelIntLvEn = Temp & 4;
            DctGangEn = Temp & 0x10;
            DctSelIntLvAddr = Temp>>6 & 3;
            DctSelBaseAddr = Temp & 0xFFFFF800;
            DctSelBaseOffsetLong = (Get_PCI(bus0, dev24 + NodeID, func2, 0x114) &
                0xFFFFFC00)<<16;

            //Determine if High range is selected
            if(DctSelHiRngEn && DctGangEn==0 && (SystemAddr>>27) >=
                (DctSelBaseAddr>>11)) HiRangeSelected = 1;
            else HiRangeSelected=0;
            //Determine Channel
            if(DctGangEn) ChannelSelect = 0;

```



```

else if (HiRangeSelected) ChannelSelect = DctSelHi;
else if (DctSelIntLvEn && DctSelIntLvAddr == 0)
    ChannelSelect = SystemAddr>>6 & 1;
else if (DctSelIntLvEn && DctSelIntLvAddr>>1 & 1)
{
    Temp = fUnaryXOR(SystemAddr>>16&0x1F); //function returns odd parity
    //1= number of set bits in argument is odd.
    //0= number of set bits in argument is even.
    if(DctSelIntLvAddr & 1) ChannelSelect = (SystemAddr>>9 & 1)^Temp;
    else ChannelSelect = (SystemAddr>>6 & 1)^Temp;
}
else if (DctSelIntLvEn && IntlvEn&4) ChannelSelect = SystemAddr>>15&1;
else if (DctSelIntLvEn && IntlvEn&2) ChannelSelect = SystemAddr>>14&1;
else if (DctSelIntLvEn && IntlvEn&1) ChannelSelect = SystemAddr>>13&1;
else if (DctSelIntLvEn) ChannelSelect = SystemAddr>>12&1;
else if (DctSelHiRngEn && DctGangEn==0) ChannelSelect = ~DctSelHi&1;
else ChannelSelect = 0;
//Determine Base address Offset to use
if(HiRangeSelected)
{
    if(!(DctSelBaseAddr & 0xFFFF0000) && (HoleEn & 1) &&
        (SystemAddr >= 0x1_00000000))
        ChannelOffsetLong = HoleOffset<<16;
    else
        ChannelOffsetLong= DctSelBaseOffsetLong;
}
else
{
    if((HoleEn & 1) && (SystemAddr >= 0x1_00000000))
        ChannelOffsetLong = HoleOffset<<16;
    else
        ChannelOffsetLong = DramBaseLong & 0xFFFF_F8000000;
}
//Remove hoisting offset and normalize to DRAM bus addresses
ChannelAddrLong = (SystemAddr & 0x0000FFFF_FFFFFFFC0) -
    (ChannelOffsetLong & 0x0000FFFF_FF800000);
//Remove Node ID (in case of processor interleaving)
Temp = ChannelAddrLong & 0xFC0;
ChannelAddrLong = (ChannelAddrLong >>Ilog & 0xFFFF_FFFFF00) | Temp;
//Remove Channel interleave and hash
if(DctSelIntLvEn && HiRangeSelected==0 && DctGangEn==0)
{
    if(DctSelIntLvAddr & 1!= 1)
        ChannelAddrLong = (ChannelAddrLong>>1) & 0xFFFFFFFF_FFFFFFFC0;
    else if(DctSelIntLvAddr == 1)
    {
        Temp = ChannelAddrLong & 0xFC0;
        ChannelAddrLong = ((ChannelAddrLong & 0xFFFFFFFF_FFFFE000) >> 1) | Temp;
    }
    else
    {
        Temp = ChannelAddrLong & 0x1C0;
        ChannelAddrLong = ((ChannelAddrLong & 0xFFFFFFFF_FFFFC00) >> 1) | Temp;
    }
}
}
InputAddr = ChannelAddrLong>>8;
for(CS = 0; CS < 8; CS++)
{
    F2Offset = 0x40 + (CS << 2);
    if ((CS % 2) == 0)

```



- In ganged mode, a 64-byte request is mapped to each of the four sequential data beats across both channels as QW0 (channel A), QW1 (channel B), QW2 (channel A)...QW7 (channel B).
- In ganged mode, two sequential 64-byte requests that are combined to form a single 128-byte request are mapped to each of the eight sequential data beats across both channels as QW0 (channel A), QW1 (channel B), QW2 (channel A)...QW15 (channel B).

## 2.8.9 DCT/DRAM Initialization

DRAM initialization involves several steps in order to configure the DRAM controllers and the DRAM, and to tune the DRAM channel for optimal performance. The following describes an ordered sequence of steps needed to accomplish setting up the memory channels from reset.

After cold reset, BIOS performs the following in order:

1. Configure the DDR supply voltage regulator. See [2.8.9.1](#).
2. Phy and controller mode configuration. See [2.8.9.2](#).
3. Phy compensation initialization. See [2.8.9.3](#).
4. DRAM controller and device initialization. See the note at the end of this section.
  - A. Program SPD timings. See [2.8.9.4](#).
  - B. Program Non-SPD timings. See [2.8.9.5](#) and all sub-sections.
  - C. DRAM device initialization. See [2.8.9.6](#).
5. Phy Fence programming. See [2.8.9.7](#).
6. For DDR3 memory, DRAM Write Leveling. See [2.8.9.9.1](#).
7. Perform second pass of steps 3 through 5 at the target frequency, if applicable. See note [2](#) at the end of this section.
8. DRAM data training. See [2.8.9.9](#) and all sub-sections:
  - A. Receiver Enable Training. See [2.8.9.9.2](#).
  - B. DQS Position Training. See [2.8.9.9.3](#).
  - C. MaxRdLat Training. See [2.8.9.9.5.1](#).
9. Program Non-SPD timings to optimal values. See [2.8.9.5](#) and all sub-sections.
10. The memory subsystem is ready for use.

Notes:

1. If both controllers are enabled and unganged then BIOS must take extra steps to ensure that the MEMCLK frequency of DCT1 is the same as the MEMCLK frequency of DCT0. See [2.8.9.8](#) for more details.
2. If DDR3 memory is used and the target frequency is not the minimum frequency supported by the controller (either 333 MHz or 400 MHz; see [F2x\[1, 0\]94\[MemClkFreq\]](#)), then BIOS must perform two initialization passes for certain steps during the DCT/DRAM initialization; the first pass BIOS configures MEMCLK to the minimum supported frequency, and the second pass BIOS configures MEMCLK frequency to the target frequency.

### 2.8.9.1 DDR3 Supply Voltage Configuration

Some processor revisions support JEDEC defined 1.5V and 1.35V DDR3 devices. See [Table 2](#). Platforms that support 1.35V operation should power on VDDIO at 1.35V until operating voltage is determined by reading the SPD ROM of all the DIMMs. BIOS should not operate DIMMs at voltages higher than supported as indicated by the SPD.

The recommended BIOS configuration sequence is as follows:

1. BIOS reads the SPD ROM of all DIMMs to determine the common operating voltages.
2. BIOS configures VDDIO to match the lowest common supported voltage based on the SPD values. See platform specific documentation for changing the voltage.
3. Additional derating of the DDR speed may be necessary. See [Table 2](#) for revision specific maximum speed. See [2.8.9.5.8 \[DRAM Address Timing and Output Driver Compensation Control\]](#) for information on voltage and electrical load specific maximum speed.
4. For S1g4 and ASB2 processors, configure VDDR based on the target frequency.
  - For 1333 MT/s, program VDDR=1.05V.
  - For 1067 MT/s or slower, program VDDR=900mV.

### 2.8.9.2 Phy and Controller Mode Configuration

To enable subsequent phy and controller register accesses to be routed correctly, BIOS must do the following:

- Program [F2x\[1, 0\]94\[LegacyBiosMode\] = 0](#).
- Program [F2x\[1, 0\]94\[Ddr3Mode\]](#), based on the platform and DIMM type.

### 2.8.9.3 Phy compensation initialization

Each normalized driver strength code in [F2x\[1, 0\]9C\\_x00\[DataDrvStren, AddrCmdDrvStren\]](#) has a corresponding D3CMP predriver calibration code that must be programmed into [F2x9C\\_x0A](#). BIOS is required to program [F2x9C\\_x0A](#) after any processor or memory reset before any DDR commands are sent to the DRAM. BIOS initializes the DDR phy compensation logic registers [F2x9C\\_x09](#) and [F2x9C\\_x0A](#) by performing the following steps:

1. BIOS disables the phy compensation register by programming [F2x\[1, 0\]9C\\_x08\[DisAutoComp\]=1](#).
2. BIOS waits 5 us for the disabling of the compensation engine to complete.
3. For each normalized driver strength code read from [F2x\[1, 0\]9C\\_x00\[AddrCmdDrvStren\]](#), program the corresponding 3 bit predriver code in [F2x9C\\_x0A\[D3Cmp1NCal, D3Cmp1PCal\]](#).
4. For each normalized driver strength code read from [F2x\[1, 0\]9C\\_x00\[DataDrvStren\]](#), program the corresponding 3 bit predriver code in [F2x9C\\_x0A\[D3Cmp0NCal, D3Cmp0PCal, D3Cmp2NCal, D3Cmp2PCal\]](#). Configurations with both channels running in ungang mode with four DIMMs at DDR533 should program 000b in [F2x9C\\_x0A\[D3Cmp0NCal, D3Cmp0PCal, D3Cmp2NCal, D3Cmp2PCal\]](#).

BIOS re-enables the phy compensation engine when DRAM initialization is complete. See [2.8.9.6](#).

### 2.8.9.4 SPD ROM-Based Configuration

The 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 the IO hub, which obtains the information through a secondary device connected to the IO hub through the SMBus. This secondary device communicates with the DIMM by means of the I<sup>2</sup>C bus. BIOS must determine the type of DRAM used on the DIMM in order to interpret the SPD byte values correctly as they differ greatly between DRR2 and DDR3. This information is available in byte 2 of the SPD in all DIMM devices.

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:

- T<sub>cl</sub>: (CAS latency)
- T<sub>rc</sub>: Active-to-Active/Auto Refresh command period
- T<sub>r<sub>fc</sub></sub>: Auto-Refresh-to-Active/Auto Refresh command period
- T<sub>rcd</sub>: Active-to-Read-or-Write delay
- T<sub>rrd</sub>: Active-Bank-A to-Active-Bank-B delay
- T<sub>ras</sub>: Active-to-Precharge delay
- T<sub>rp</sub>: Precharge time
- T<sub>ref</sub>: Refresh interval
- T<sub>trp</sub>: Internal Read to Precharge command delay
- T<sub>wtr</sub>: Internal Write to Read command delay
- T<sub>wr</sub>: Write recovery time

Optimal cycle time is specified for each DIMM and is used to limit or determine bus frequency. See [2.8.9.8 \[DRAM Channel Frequency Change\]](#) for more information on configuring the bus frequency.

BIOS should follow the JEDEC SPD specification to determine the frequency and CAS latency supported by each DIMM. BIOS should limit the frequency based on processor support. See [2.8.9.5.8 \[DRAM Address Timing and Output Driver Compensation Control\]](#). If a channel has multiple DIMMs with multiple supported CAS latencies, BIOS should choose the highest processor supported frequency supported by all DIMMs on the channel for which a common supported CAS latency exists. If BIOS does not find a common supported CAS latency, then BIOS should revert to a fail-safe frequency and CAS latency that all of the DRAMs are capable of supporting as defined by the JEDEC DRAM specification.

### 2.8.9.5 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:

- DDR2 vs. DDR3 DRAM types.
- Mixed or non-mixed DIMMs (x4 with x8).
- Training delay values. See [2.8.9.9 \[DRAM Training\]](#).
- Read and write latency differences.
- The phy's idle clock requirements on the data bus.
- DDR3 ODT timing requirements.

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.

#### 2.8.9.5.1 Trdrd (Read to Read Timing)

The optimal value for `F2x[1, 0]8C[Trdrd]` 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 value after DDR training is complete:

- BIOS calculates Trdrd (in MEMCLKs) =  $CGDD_{Trdrd} / 2 + 3$  clocks and programs `F2x[1, 0]8C[Trdrd]` with

the encoded field value. BIOS rounds fractional values down. BIOS has two options for calculating CGDD:

- The conservative Critical Gross Delay Difference for Trdrd ( $CGDD_{Trdrd}$ ) is the largest  $F2x[1, 0]9C\_x[2B:10][DqsRcvEnGrossDelay]$  delay of all byte lanes and all DIMMs minus the smallest  $F2x[1, 0]9C\_x[2B:10][DqsRcvEnGrossDelay]$  delay of all byte lanes and all DIMMs.
- The optimal Critical Gross Delay Difference (in UI) for Trdrd ( $CGDD_{Trdrd}$ ) is the largest GDD of all byte lanes. The GDD on each byte lane is the largest  $F2x[1, 0]9C\_x[2B:10][DqsRcvEnGrossDelay]$  delay of that byte lane of any DIMM minus the smallest  $F2x[1, 0]9C\_x[2B:10][DqsRcvEnGrossDelay]$  delay of that byte lane of any other DIMM.

The generalized optimal CGDD (in UI) for Trdrd, Twrwr, Twrrd, and TrwtTO is implemented in software as follows:

```
(signed int)CGDD = (signed int)GDD = 0;
for(i = 0; i < MAX_DIMMS; i++)
{
    for(j = 0; j < MAX_DIMMS; j++)
    {
        for(k = 0; k < MAX_BYTELANES; k++)
        {
            GDD = Delay_Term1[i][k] - Delay_Term2[j][k];
            if (GDD > CGDD) CGDD = GDD;
        }
    }
}
```

#### 2.8.9.5.2 Twrwr (Write to Write Timing)

The optimal value for  $F2x[1, 0]8C[Twrwr]$  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; otherwise, BIOS should program Twrwr as follows:

- BIOS calculates  $Twrwr$  (in MEMCLKs) =  $CGDD_{Twrwr} / 2 + 3$  clocks and programs  $F2x[1, 0]8C[Twrwr]$  with the encoded field value. BIOS rounds fractional values down. BIOS has two options for calculating CGDD:
  - The conservative Critical Gross Delay Difference for Twrwr ( $CGDD_{Twrwr}$ ) is the largest  $F2x[1, 0]9C\_x[3:0]0[2:1]:F2x[1, 0]9C\_x[3:0]03[WrDatGrossDlyByte]$  delay of all byte lanes and all DIMMs minus the smallest  $F2x[1, 0]9C\_x[3:0]0[2:1]:F2x[1, 0]9C\_x[3:0]03[WrDatGrossDlyByte]$  delay of all byte lanes and all DIMMs.
  - The optimal Critical Gross Delay Difference (in UI) for Twrwr ( $CGDD_{Twrwr}$ ) is the largest GDD of all byte lanes. The GDD on each byte lane is the largest  $F2x[1, 0]9C\_x[3:0]0[2:1]:F2x[1, 0]9C\_x[3:0]03[WrDatGrossDlyByte]$  delay of that byte lane of any DIMM minus the smallest  $F2x[1, 0]9C\_x[3:0]0[2:1]:F2x[1, 0]9C\_x[3:0]03[WrDatGrossDlyByte]$  delay of that byte lane of any other DIMM. See “generalized optimal CGDD” in [The Trdrd (Read to Read Timing)] 2.8.9.5.1.

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

The optimal value for  $F2x[1, 0]8C[Twrrd]$  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; otherwise, BIOS should use the guidelines below to configure the recommended platform generic timing value after DDR training is complete:

- BIOS calculates  $Twrrd$  (in MEMCLKs) =  $CGDD_{Twrrd} / 2 - LD + 3$  clocks and programs  $F2x[1, 0]8C[Twrrd]$  with the encoded field value. BIOS rounds fractional values down.

- For DDR3, BIOS calculates the latency difference (LD) as equal to read CAS latency minus write CAS latency, in MEMCLKs (see  $F2x[1, 0]88[Te]$  and  $F2x[1, 0]84[Tcw]$ ) which can be a negative or positive value.
- For DDR2, the LD is always one clock.
- BIOS has two options for calculating CGDD:
  - The Critical Gross Delay Difference for  $T_{wrrd}$  ( $CGDD_{T_{wrrd}}$ ) is the largest  $F2x[1, 0]9C\_x[3:0]0[2:1]:F2x[1, 0]9C\_x[3:0]03[WrDatGrossDlyByte]$  delay of all byte lanes and all DIMMs minus the smallest  $F2x[1, 0]9C\_x[2B:10][DqsRcvEnGrossDelay]$  delay of all byte lanes and all DIMMs.
  - The optimal Critical Gross Delay Difference (in UI) for  $T_{wrrd}$  ( $CGDD_{T_{wrrd}}$ ) is the largest GDD of all byte lanes. The GDD on each byte lane is the largest  $F2x[1, 0]9C\_x[3:0]0[2:1]:F2x[1, 0]9C\_x[3:0]03[WrDatGrossDlyByte]$  delay of that byte lane of any DIMM minus the smallest  $F2x[1, 0]9C\_x[2B:10][DqsRcvEnGrossDelay]$  delay of that byte lane of any other DIMM. See “generalized optimal CGDD” in [The  $T_{rdrd}$  (Read to Read Timing)] 2.8.9.5.1.

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

The optimal value for  $F2x[1, 0]8C[TrwtTO]$  is platform and configuration specific and should be characterized for best performance. Prior to DRAM training, BIOS should program this parameter to the largest defined value; otherwise, BIOS should use the guidelines below to configure the recommended platform generic timing value after DDR training is complete:

- BIOS calculates  $TrwtTO$  (in MEMCLKs) =  $CGDD_{TrwtTO} / 2 + LD + 3$  clocks and programs  $F2x[1, 0]8C[TrwtTO]$  with the converted field value. BIOS rounds fractional values down.
  - For DDR3, BIOS calculates the latency difference (LD) as equal to read CAS latency minus write CAS latency, in MEMCLKs (see  $F2x[1, 0]88[Te]$  and  $F2x[1, 0]84[Tcw]$ ) which can be a negative or positive value.
  - For DDR2, the LD is always one clock.
- BIOS has two options for calculating CGDD:
  - The Critical Gross Delay Difference for  $TrwtTO$  ( $CGDD_{TrwtTO}$ ) is the largest  $F2x[1, 0]9C\_x[2B:10][DqsRcvEnGrossDelay]$  delay of all byte lanes and all DIMMs minus the smallest  $F2x[1, 0]9C\_x[3:0]0[2:1]:F2x[1, 0]9C\_x[3:0]03[WrDatGrossDlyByte]$  delay of all byte lanes and all DIMMs.
  - The optimal Critical Gross Delay Difference (in UI) for  $TrwtTO$  ( $CGDD_{TrwtTO}$ ) is the largest GDD of all byte lanes. The GDD on each byte lane is the largest  $F2x[1, 0]9C\_x[2B:10][DqsRcvEnGrossDelay]$  delay of that byte lane of any DIMM minus the smallest  $F2x[1, 0]9C\_x[3:0]0[2:1]:F2x[1, 0]9C\_x[3:0]03[WrDatGrossDlyByte]$  delay of that byte lane of any other DIMM. See “generalized optimal CGDD” in [The  $T_{rdrd}$  (Read to Read Timing)] 2.8.9.5.1.

#### 2.8.9.5.5 TrwtWB (Read-to-Write Turnaround for Opportunistic Write Bursting)

This timing parameter,  $F2x[1, 0]8C[TrwtWB]$ , ensures read-to-write data-bus turnaround. This value should be one more than the programmed  $F2x[1, 0]8C[TrwtTO]$  value. See [The  $T_{rwtTO}$  (Read-to-Write Turnaround for Data, DQS Contention)] 2.8.9.5.4.

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

No more than 4 banks may be activated in a rolling tFAW window. For DDR2 devices, BIOS must convert the tFAW parameter into MEMCLK cycles by dividing the highest tFAW parameter (in ns) found in all the SPD ROMs for DIMMs connected to the channel by the period of MEMCLK (in ns) and rounding up to the next integer. For example, if this field is set to 10 clocks and an activate command is issued in clock N, then no more than three further activate commands may be issued in clocks N+1 through N+9. Table 37 shows the



DDR2 F2x[1, 0]94[tFAW] clock values used for various frequencies and page sizes.

**Table 37: DDR2 Four Bank Activate Window Values**

Page Size	533 MHz	400 MHz	333 MHz	266 MHz	200 MHz
1K	19 MEMCLKs	14 MEMCLKs	13 MEMCLKs	10 MEMCLKs	8 MEMCLKs
2K	24 MEMCLKs	18 MEMCLKs	17 MEMCLKs	14 MEMCLKs	10 MEMCLKs

For DDR3, BIOS should use the tFAW values specified in the SPD ROM for the specific DIMM device.

### 2.8.9.5.7 DRAM ODT Control

This section describes the ODT configurations and settings for the processor and attached DIMMs. The tables specify ODT values for different speeds and configurations, on a per channel basis. The processor ODT values are controlled by F2x[1, 0]9C\_x00[ProcOdt] for both DDR2 and DDR3. The DIMM termination values are programmed as specified below before DRAM device initialization. If the DIMM termination values are changed after device initialization then BIOS must issue MRS commands to the devices to change the values. See F2x[1, 0]7C for more information.

Table 38 and Table 39 document the ODT termination values for different DDR2 configurations. The DDR2 DIMM nominal termination resistance is controlled by F2x[1, 0]90[DramTerm].

Table 42 documents the ODT nominal (non-write) and dynamic termination resistance values for different DDR3 unbuffered DIMM configurations. The DDR3 DIMM nominal termination resistance is controlled by F2x[1, 0]84[DramTerm]. The DDR3 DIMM dynamic termination resistance is controlled by F2x[1, 0]84[DramTermDyn].

Table 43 and Table 44 document the ODT nominal (non-write) and dynamic termination resistance values for different DDR3 registered DIMM configurations. The BIOS enables nominal termination on even numbered ranks and disables nominal termination of odd numbered ranks of a quad rank DIMM. In addition, BIOS configures different ODT values for single rank or dual rank DIMMs on a channel including quad rank DIMMs, as specified in this table.

**Table 38: ODT Settings for DDR2 SO-DIMMs**

DDR Type-Rate	Number of DIMMs	Processor ODT	DIMM ODT
533, 667, 800	1	150 ohms	150 ohms
533, 667	2	150 ohms	75 ohms
800	2	150 ohms	50 ohms

**Table 39: ODT Settings for DDR2 UDIMMs and RDIMMs**

DDR Rate	Number of DIMMs	Processor ODT	DIMM ODT
400, 533, 667	1	75 ohms	75 ohms
400, 533	2 or more	150 ohms	75 ohms
667	2	150 ohms	75 ohms
800	1	75 ohms	75 ohms
800	2	150 ohms	50 ohms
1066	1	75 ohms	75 ohms

**Table 40: ODT Settings for DDR3 SO-DIMMs (1 DIMM per channel)**

DDR Rate	Number of DIMMs	Processor ODT	DIMM ODT (Rtt_Nom)	DIMM Dynamic ODT (Rtt_Wr)
800	1	240 ohms	120 ohms	Disabled
1066	1	120 ohms	60 ohms	Disabled
1333	1	60 ohms	60 ohms	Disabled

**Table 41: ODT Setting for DDR3 SO-DIMMs (2 DIMMs per channel)**

DDR Rate	Number of DIMMs	Processor ODT	DIMM ODT (Rtt_Nom)	DIMM Dynamic ODT (Rtt_Wr)
800 - 1066	1	60 ohms	60 ohms	Disabled
800	2	60 ohms	40 ohms	120 ohms
1066		60 ohms	30 ohms	120 ohms

**Table 42: ODT Settings for DDR3 UDIMMs (2 or 3 DIMMs per channel)**

DDR Rate	Number of DIMMs	Processor ODT	DIMM ODT (Rtt_Nom)	DIMM Dynamic ODT (Rtt_Wr)
800 - 1333	1 SR or DR DIMM on 2 DIMM/Ch platform or 1 SR DIMM on 3 DIMM/Ch platform	60 ohms	60 ohms	Disabled
800 - 1333	1 DR DIMM on 3 DIMM/Ch platform	60 ohms	Rank0: 60 ohms Rank1: Disabled	60 ohms
800 - 1066	2	60 ohms	40 ohms	120 ohms
1333		60 ohms	30 ohms	120 ohms

**Table 43: ODT Settings for DDR3 RDIMMs (2 DIMMs per channel)**

DDR Rate	Number of DIMMs <sup>1</sup>	Processor ODT	SR or DR DIMM ODT (Rtt_Nom)	QR DIMM ODT (Rtt_Nom) <sup>2</sup>	DIMM Dynamic ODT (Rtt_Wr)
800 - 1333	1 SR or DR	60 ohms	60 ohms	-	Disabled
800 - 1066	2 SR or DR	60 ohms	40 ohms	-	120 ohms
1333		60 ohms	30 ohms	-	120 ohms
800 - 1066	1 QR	60 ohms	-	60 ohms	120 ohms
800	1 SR or DR and 1 QR	60 ohms	30 ohms	60 ohms	120 ohms
1066		60 ohms	30 ohms	60 ohms	60 ohms

**Table 43: ODT Settings for DDR3 RDIMMs (2 DIMMs per channel)**

DDR Rate	Number of DIMMs <sup>1</sup>	Processor ODT	SR or DR DIMM ODT (Rtt_Nom)	QR DIMM ODT (Rtt_Nom) <sup>2</sup>	DIMM Dynamic ODT (Rtt_Wr)
800	2 QR	60 ohms	-	60 ohms	120 ohms
1066	2 QR	60 ohms	-	60 ohms	60 ohms

1. SR = Single Rank, DR = Dual Rank, QR = Quad Rank.  
2. BIOS enables Rtt\_Nom only on ranks S0 and S2 of a QR DIMM.

**Table 44: ODT Settings for DDR3 RDIMMs (3 DIMMs per channel)**

DDR Rate	Number of DIMMs <sup>1</sup>	Processor ODT	SR or DR DIMM ODT <sup>2</sup> (Rtt_Nom)	QR DIMM ODT <sup>2</sup> (Rtt_Nom)	DIMM Dynamic ODT (Rtt_Wr)
800 - 1333	1 SR or DR	60 ohms	60 ohms	-	Disabled
800 - 1066	2 SR or DR	60 ohms	40 ohms	-	120 ohms
1333		60 ohms	30 ohms	-	120 ohms
800 - 1333	3 SR or DR	60 ohms	30 or 40 ohms <sup>3</sup>	-	120 ohms
800 - 1066	1 QR	60 ohms	-	60 ohms	120 ohms
800	1 SR or DR and 1 QR	60 ohms	30 ohms	60 ohms	120 ohms
1066		60 ohms	30 ohms	60 ohms	60 ohms
800	2 SR or DR and 1 QR	60 ohms	40 ohms	60 ohms	120 ohms

1. SR = Single Rank, DR = Dual Rank, QR = Quad Rank.  
2. BIOS enables Rtt\_Nom only on rank S0 of a DR DIMM and ranks S0 and S2 of a QR DIMM.  
3. If C32 package with 3 single rank DIMMs at DDR-1066 at 1.5V or if G34 package (any three SR or DR DIMMs) then Rtt\_Nom=30 ohms; else Rtt\_Nom=40 ohms.

The following describes the general ODT behavior for various DDR2 system configurations. In all cases, the processor ODT is off for writes and is on for reads:

- For 1 DIMM on a channel:
  - For writes, the first rank of the DIMM provides ODT.
  - For reads, the DIMM ODT is off for all ranks.
- For 2 DIMMs on a channel:
  - For writes and reads:
    - ODT is on for the first rank of the non-targeted DIMM.
    - ODTs are off for all other ranks on the channel.
- For more than 2 DIMMs on a channel:
  - For writes and reads:
    - ODT is active for the first rank of all the non-target DIMMs. The target DIMM being written has ODT turned off for all ranks.

The following describes the general ODT behavior for various DDR3 system configurations. In all cases, the processor ODT is off for writes and is on for reads:

- For one dual rank DIMM on a channel:
  - For writes, the ODT of the target rank is off and the non-target rank is on. If the DIMM is a single rank DIMM, then ODT is on for that rank.

- For reads, the DIMM ODT is off for all ranks.
- For two dual rank DIMMs on a channel:
  - For writes, the ODT is on for the target rank of the target DIMM and also on for the first rank of the non-target DIMM.
  - For reads, ODT is on for the first rank of the non-target DIMM.
- For one 4 rank DIMM on a channel:
  - For writes to any even target rank (i.e. ranks 0 or 2), the ODT is on for all even ranks. For writes to rank 1, the ODT is on for rank 1 and for rank 2. For writes to rank 3, the ODT is on for rank 3 and rank 0.
  - For reads, the DIMM ODT is off for all ranks.
- For one single rank DIMM and one 4 rank DIMM on a channel:
  - For writes, if the target is the single rank DIMM, then ODT is on for the target rank and the ODT is on for all even ranks of the 4 rank DIMM. ODT is off for all other ranks.
  - For writes, if the target rank is on the 4 rank DIMM, then ODT is on for the single rank DIMM and ODT is on for the target rank of the 4 rank DIMM. ODT is off for all other ranks.
  - For reads, if the target is the single rank DIMM, then ODT is on for all even ranks of the 4 rank DIMM only. ODT is off for all other ranks.
  - For reads, if the target rank is on the 4 rank DIMM, then ODT is on for the single rank DIMM only.
- For one dual rank DIMM and one 4 rank DIMM on a channel:
  - For writes, if the write is to one of the ranks on the target dual rank DIMM, then ODT is on for the target rank and ODT is also on for all even ranks of the 4 rank DIMM. ODT is off for all other ranks.
  - For writes, if the write is to a rank on the 4 rank DIMM, then ODT is on for the first rank of the dual rank DIMM and ODT is also on for the target rank of the 4 rank DIMM. ODT is off for all other ranks.
  - For reads, if the read is from one of the ranks of the dual rank DIMM, then ODT is on for all even ranks of the non-target 4 rank DIMM.
  - For reads, if the read is from a rank on the 4 rank DIMM, then ODT is on for the first rank of the dual rank DIMM only.
- For two 4 rank DIMMs on a channel:
  - For writes, if the write is to one of the ranks of a 4 rank DIMM, then ODT is on for that target rank and ODT is also on for all even ranks of the non-target 4 rank DIMM. ODT is off for all other ranks.
  - For reads, if the read is from one of the ranks of a 4 rank DIMM, then ODT is on for all even ranks of the non-target 4 rank DIMM.
- For more than two registered DIMMs on a channel:
  - For writes, ODT is on for the first rank of all the DIMMs.
  - For reads, ODT is on for the first rank of all the non-target DIMMs. ODT is off for the target DIMM.

On revision D and later processors, the BIOS may additionally configure the ODT pattern by setting `F2x[1, 0]A8[ProgOdtEn]`. The ODT patterns for reads and writes are programmed using `F2x[1, 0]F4_x[181:180]` and `F2x[1, 0]F4_x[183:182]`, respectively, as specified by the tables below for a given platform type and DIMM population. Table 45, Table 47, Table 49, and Table 49 document the recommended ODT patterns for each DRAM channel.

For DDR3 system configurations, the BIOS can configure the ODT pattern write leveling by setting `F2x[1, 0]9C_x08[WrlvOdtEn]` and programming the DIMM ODT pattern using `F2x[1, 0]9C_x08[WrlvOdt[3:0]]`. Table 46, Table 48, and Table 50 document the recommended ODT patterns for each target DIMM. See 2.8.9.9.1 [Write Levelization Training].

**Table 45: DDR3 UDIMM ODT Pattern**

DIMM0	DIMM1	F2x[1, 0]F4_x[181:180]		F2x[1, 0]F4_x[183:182]	
		x181	x180	x183	x182
-	SR	0000_0000h	0000_0000h	0000_0000h	0002_0000h
-	DR	0000_0000h	0000_0000h	0000_0000h	0208_0000h
SR/DR	SR/DR	0000_0000h	0101_0202h	0000_0000h	0903_0603h

**Table 46: DDR3 SO-DIMM and UDIMM Write Leveling ODT Pattern**

DIMM0	DIMM1	F2x[1, 0]9C_x08[WrlvOdt[3:0]]		
		2 DIMM slots per channel <sup>2</sup>		1 DIMM slot per channel <sup>3</sup>
		DIMM0 <sup>1</sup>	DIMM1 <sup>1</sup>	DIMM0 <sup>1</sup>
-	SR	-	2h	N/A
-	DR	-	8h	N/A
SR/DR	SR/DR	3h	3h	N/A
SR	N/A	N/A	N/A	1h
DR	N/A	N/A	N/A	4h

1. Target DIMM.  
 2. For Unbuffered DIMMs.  
 3. For SO-DIMMs.

**Table 47: DDR3 RDIMM ODT Pattern (2 DIMMs per channel)**

DIMM0 <sup>4</sup>	DIMM1 <sup>4</sup>	F2x[1, 0]F4_x[181:180]		F2x[1, 0]F4_x[183:182]	
		x181	x180	x183	x182
-	SR	0000_0000h	0000_0000h	0000_0000h	0002_0000h
-	DR	0000_0000h	0000_0000h	0000_0000h	0208_0000h
-	QR	0000_0000h	0000_0000h	0A0A_0000h	0A0A_0000h
SR/DR	SR/DR	0000_0000h	0101_0202h	0000_0000h	0903_0603h
SR/DR	QR	0101_0000h	0101_0A0Ah	0109_0000h	0103_0E0Bh
QR	SR/DR	0000_0202h	0505_0202h	0000_0206h	0D07_0203h
QR	QR	0505_0A0Ah	0505_0A0Ah	050D_0A0Eh	0507_0A0Bh

4. Daisy chain topology.

**Table 48: DDR3 RDIMM Write Leveling ODT Pattern (2 DIMMs per channel)**

DIMM0	DIMM1	F2x[1, 0]9C_x08[WrlvOdt[3:0]]	
		DIMM0 <sup>1</sup>	DIMM1 <sup>1</sup>
-	SR	-	2h
-	DR	-	8h
-	QR	-	Ah/Ah
SR/DR	SR/DR	3h	3h

1. Target DIMM. QR represented as target ranks S0/S2.

**Table 48: DDR3 RDIMM Write Leveling ODT Pattern (2 DIMMs per channel)**

DIMM0	DIMM1	F2x[1, 0]9C_x08[WrlvOdt[3:0]]	
		DIMM0 <sup>1</sup>	DIMM1 <sup>1</sup>
SR/DR	QR	Bh	3h/9h
QR	SR/DR	3h/6h	7h
QR	QR	Bh/Eh	7h/Dh

1. Target DIMM. QR represented as target ranks S0/S2.

**Table 49: DDR3 UDIMM and RDIMM ODT Pattern (3 DIMMs per channel)**

DIMM0	DIMM1 <sup>1</sup>	DIMM2	F2x[1, 0]F4_x[181:180]		F2x[1, 0]F4_x[183:182]	
			x181	x180	x183	x182
-	-	SR/DR	0000_0000h	0000_0000h	0000_0404h	0000_0000h
SR/DR	-	SR/DR	0000_0101h	0000_0404h	0000_0105h	0000_0405h
SR/DR	SR/DR	SR/DR	0000_0303h	0505_0606h	0000_0307h	0D07_0607h
-	QR	-	0000_0000h	0000_0000h	020A_0000h	080A_0000h
-	QR	SR/DR	0404_0A0Ah	0404_0000h	040C_0A0Eh	0406_0000h
SR/DR	QR	SR/DR	0505_0B0Bh	0505_0E0Eh	050D_0B0Fh	0507_0E0Fh

1. This slot does not support an unbuffered DIMM.

**Table 50: DDR3 UDIMM and RDIMM Write Leveling ODT Pattern (3 DIMM per channel)**

DIMM0	DIMM1 <sup>1</sup>	DIMM2	F2x[1, 0]9C_x08[WrlvOdt[3:0]]		
			DIMM0 <sup>2</sup>	DIMM1 <sup>2</sup>	DIMM2 <sup>2</sup>
-	-	SR/DR	-	-	4h
SR/DR	-	SR/DR	5h	-	5h
SR/DR	SR/DR	SR/DR	7h	7h	7h
-	QR	-	-	Ah/Ah	-
-	QR	SR/DR	-	6h/Ch	Eh
SR/DR	QR	SR/DR	Fh	7h/Dh	Fh

1. This slot does not support an unbuffered DIMM.  
2. Target DIMM. QR represented as target ranks S0/S2.

### 2.8.9.5.8 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 tables below document the address timing and output driver settings on a per channel basis for different DDR DIMM types. The DIMMs on each channel are numbered from 0 to n where DIMM0 is the DIMM closest to the processor on that channel and DIMMn is the DIMM farthest from the processor on that channel. DIMMs must be populated from farthest slot to closest slot to the processor on a per channel basis. Populations that are not shown in these tables are not supported. These tables document the optimal settings for motherboards which meet the relevant motherboard design guidelines. See [2.8 \[DRAM Controllers \(DCTs\)\]](#) for an overview of the DIMM and memory bus speed support.

**Table 51: DDR2 SO-DIMM Address Timings and Drive Strengths for 1 DIMM per channel**

DDR Type-Rate	DIMM0	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>1</sup>
DDR2-533	SRx16	1T	002F_2F2Fh	X011_1222h
	SRx8			
	DRx16		002C_2C2Ch	
	DRx8			
DDR2-667	SRx16	1T	0027_2727h	X011_1222h
	SRx8		002A_2A2Ah	
	DRx16			
	DRx8	2T	0000_2828h	
DDR2-800	SRx16	1T	0029_2929h	X011_1222h
	SRx8		002A_2A2Ah	
	DRx16			
	DRx8	2T	0000_2F2Fh	

1. See [Table 38](#).

**Table 52: DDR2 SO-DIMM Address Timings and Drive Strengths for 2 DIMMs per channel**

DDR Type-Rate	DIMM0 <sup>1</sup>	DIMM1 <sup>1</sup>	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>2</sup>
DDR2-533	-	SRx16	1T	002B_2F00h	X011_1222h
	-	DRx8			
DDR2-533	-	SRx8	1T	002F_2F00h	X011_1222h
DDR2-533	SRx16	SRx16	2T	002F_2F00h	X011_1322h
	SRx16	SRx8			
	SRx8	SRx16			
DDR2-533	SRx8	SRx8	2T	0000_2F00h	X011_1322h
DDR2-533	DRx8	DRx8	2T	0034_2F00h	X011_1322h
DDR2-533	DRx8	SRx16	2T	0038_2F00h	X011_1322h
	SRx16	DRx8			
DDR2-533	DRx8	SRx8	2T	0037_2F00h	X011_1322h
	SRx8	DRx8			
DDR2-667	-	any	1T	0020_2220h	X011_1222h
DDR2-667	SRx16	SRx16	2T	0020_2220h	X011_1322h
	SRx16	SRx8			
	SRx8	SRx16			
DDR2-667	SRx8	SRx8	2T	0030_2220h	X011_1322h
DDR2-667	DRx8	DRx8	2T	002B_2220h	X011_1322h
DDR2-667	DRx8	SRx16	2T	002C_2220h	X011_1322h
	SRx16	DRx8			



**Table 52: DDR2 SO-DIMM Address Timings and Drive Strengths for 2 DIMMs per channel**

DDR Type-Rate	DIMM0 <sup>1</sup>	DIMM1 <sup>1</sup>	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>2</sup>
DDR2-667	DRx8	SRx8	2T	002A_2220h	X011_1322h
	SRx8	DRx8			
DDR2-800	-	any	2T	0020_2520h	X011_3222h
DDR2-800	any	any	2T	0020_2520h	X011_3322h

1. any = SRx8, SRx16, DRx8, or DRx16.  
 2. See [Table 39](#).

**Table 53: DDR2 UDIMM Address Timings and Drive Strengths for 2 DIMMs per channel**

DDR Type-Rate	DIMM0 <sup>1</sup>	DIMM1 <sup>1</sup>	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>2</sup>
DDR2-400	-	any	1T	002F_2F00h	X011_1222h
DDR2-400	any	any	2T	002F_2F00h	X011_1322h
DDR2-533	-	SRx16	1T	002B_2F00h	X011_1222h
	-	DRx8			
DDR2-533	-	SRx8	1T	002F_2F00h	X011_1222h
DDR2-533	SRx16	SRx16	2T	002F_2F00h	X011_1322h
	SRx16	SRx8			
	SRx8	SRx16			
DDR2-533	SRx8	SRx8	2T	0000_2F00h	X011_1322h
DDR2-533	DRx8	DRx8	2T	0034_2F00h	X011_1322h
DDR2-533	DRx8	SRx16	2T	0038_2F00h	X011_1322h
	SRx16	DRx8			
DDR2-533	DRx8	SRx8	2T	0037_2F00h	X011_1322h
	SRx8	DRx8			
DDR2-667	-	any	1T	0020_2220h	X011_1222h
DDR2-667	SRx16	SRx16	2T	0020_2220h	X011_1322h
	SRx16	SRx8			
	SRx8	SRx16			
DDR2-667	SRx8	SRx8	2T	0030_2220h	X011_1322h
DDR2-667	DRx8	DRx8	2T	002B_2220h	X011_1322h
DDR2-667	DRx8	SRx16	2T	002C_2220h	X011_1322h
	SRx16	DRx8			
DDR2-667	DRx8	SRx8	2T	002A_2220h	X011_1322h
	SRx8	DRx8			
DDR2-800	-	any	2T	0020_2520h	X011_3222h
DDR2-800	any	any	2T	0020_2520h	X011_3322h

**Table 53: DDR2 UDIMM Address Timings and Drive Strengths for 2 DIMMs per channel**

DDR Type-Rate	DIMM0 <sup>1</sup>	DIMM1 <sup>1</sup>	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>2</sup>
DDR2-1066	-	DRx8	2T	002F_2020h	X011_3000h
		SRx8			
		SRx16			

1. any = SRx8, SRx16, DRx8, or DRx16.  
 2. See [Table 39](#).

**Table 54: DDR2 RDIMM Address Timings and Drive Strengths for 4 DIMMs per channel**

DDR Type-Rate	DIMM0 <sup>1</sup>	DIMM1 <sup>1</sup>	DIMM2 <sup>1</sup>	DIMM3 <sup>1</sup>	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>2</sup>
DDR2-400	-	-	-	any	1T	0000_0000h	X011_1222h
DDR2-400	-	-	any	any	1T	0037_0000h	X011_1222h
DDR2-400	-	any	any	any	1T	002F_0000h	X011_1222h
DDR2-400	any	any	any	any	1T	002F_0000h	X011_1222h
DDR2-533	-	-	-	any	1T	0000_0000h	X011_1222h
DDR2-533	-	-	any	any	1T	0037_0000h	X011_1222h
DDR2-533	-	any	any	any	1T	002F_0000h	X011_1222h
DDR2-533	any	any	any	any	1T	002F_0000h	X011_1222h
DDR2-667	-	-	-	SR or DRx8	1T	0000_0000h	X011_1222h
DDR2-667	-	-	-	DRx4	1T	0000_2F00h	X011_1222h
DDR2-667	-	-	SR	SR	1T	0037_0000h	X011_1222h
DDR2-667	-	-	SR	DR	1T	0037_3800h	X011_1222h
DDR2-667	-	-	DR	SR	1T	0037_3800h	X011_1222h
DDR2-667	-	-	DRx8	DRx8	1T	0037_0000h	X011_1222h
DDR2-667	-	-	DRx8	DRx4	1T	0037_2F00h	X011_1222h
DDR2-667	-	-	DRx4	DRx4 or DRx8	1T	0037_2F00h	X011_1222h
DDR2-800	-	-	-	SR or DRx8	1T	0000_0000h	X011_1222h
DDR2-800	-	-	-	DRx4	1T	0000_2F00h	X011_1222h
DDR2-800	-	-	SR	SR	1T	0037_0000h	X011_1222h
DDR2-800	-	-	SR	DR	1T	0037_3800h	X011_1222h
DDR2-800	-	-	DR	SR	1T	0037_3800h	X011_1222h
DDR2-800	-	-	DRx8	DRx8	1T	0037_0000h	X011_1222h
DDR2-800	-	-	DRx8	DRx4	1T	0037_2F00h	X011_1222h
DDR2-800	-	-	DRx4	DRx4 or DRx8	1T	0037_2F00h	X011_1222h

1. any = SR or DR.  
 2. See [Table 39](#).

**Table 55: DDR2 RDIMM Address Timings and Drive Strengths for 2 DIMMs per channel**

DDR Type-Rate	DIMM0 <sup>1</sup>	DIMM1 <sup>1</sup>	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>2</sup>
DDR2-400	-	any	1T	0000_0000h	X011_1222h
DDR2-400	SR or DR	SR or DR	1T	0037_0000h	X011_1222h
DDR2-400	QR	QR	1T	002F_0000h	X011_1222h
DDR2-533	-	any	1T	0000_0000h	X011_1222h
DDR2-533	SR or DR	SR or DR	1T	0037_0000h	X011_1222h
DDR2-533	QR	QR	1T	002F_0000h	X011_1222h
DDR2-667	-	SR or DRx8	1T	0000_0000h	X011_1222h
DDR2-667	-	DRx4 or QR	1T	0000_2F00h	X011_1222h
DDR2-667	SR or DRx8	SR or DRx8	1T	0037_0000h	X011_1222h
DDR2-667	SR or DR	DRx4	1T	0037_2F00h	X011_1222h
DDR2-667	DRx4	SR or DR	1T	0037_2F00h	X011_1222h
DDR2-800	-	SR or DRx8	1T	0000_0000h	X011_1222h
DDR2-800	-	DRx4	1T	0000_2F00h	X011_1222h
DDR2-800	SR or DRx8	SR or DRx8	1T	0037_0000h	X011_1222h
DDR2-800	SR or DR	DRx4	1T	0037_2F00h	X011_1222h
DDR2-800	DRx4	SR or DR	1T	0037_2F00h	X011_1222h

1. any = SR, DR, or QR

 2. See [Table 39](#).

**Table 56: AM3 DDR3 UDIMM Address Timings and Drive Strengths**

DDR Rate 1.5V <sup>1</sup>	DDR Rate 1.35V <sup>1</sup>	DIMM0	DIMM1 or DIMM2 <sup>2</sup>	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>3</sup>
800	800	-	SR-x16	1T	0000_0000h	2011_3222h
800	800	-	SR-x8	1T	0000_0000h	2011_3222h
800	800	-	DR-x8	1T	003B_0000h	2011_3222h
800	800	SR-x16	SR-x16	1T	0039_0039h	2022_3323h
800	800	SR-x8	SR-x8	1T	0039_0039h	2022_3323h
800	800	DR-x8	DR-x8	1T	0039_0039h	2022_3323h
800	800	SR-x16	SR-x8	1T	0039_0039h	2022_3323h
800	800	SR-x8	SR-x16	1T	0039_0039h	2022_3323h
800	800	SR-x16	DR-x8	1T	0039_0039h	2022_3323h
800	800	DR-x8	SR-x16	1T	0039_0039h	2022_3323h
800	800	SR-x8	DR-x8	1T	0039_0039h	2022_3323h
800	800	DR-x8	SR-x8	1T	0039_0039h	2022_3323h
1066	1066	-	SR-x16	1T	0000_0000h	2011_3222h
1066	1066	-	SR-x8	1T	0000_0000h	2011_3222h
1066	1066	-	DR-x8	1T	0038_0000h	2011_3222h
1066	1066	SR-x16	SR-x16	1T	0035_0037h	2022_3323h

**Table 56: AM3 DDR3 UDIMM Address Timings and Drive Strengths**

DDR Rate 1.5V <sup>1</sup>	DDR Rate 1.35V <sup>1</sup>	DIMM0	DIMM1 or DIMM2 <sup>2</sup>	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>3</sup>
1066	1066	SR-x8	SR-x8	1T	0035_0037h	2022_3323h
1066	1066	DR-x8	DR-x8	1T	0035_0037h	2022_3323h
1066	1066	SR-x16	SR-x8	1T	0035_0037h	2022_3323h
1066	1066	SR-x8	SR-x16	1T	0035_0037h	2022_3323h
1066	1066	SR-x16	DR-x8	1T	0035_0037h	2022_3323h
1066	1066	DR-x8	SR-x16	1T	0035_0037h	2022_3323h
1066	1066	SR-x8	DR-x8	1T	0035_0037h	2022_3323h
1066	1066	DR-x8	SR-x8	1T	0035_0037h	2022_3323h
1333	N/A	-	SR-x16	1T	0000_0000h	2011_3222h
1333	N/A	-	SR-x8	1T	0000_0000h	2011_3222h
1333	N/A	-	DR-x8	1T	0036_0000h	2011_3222h
1333	N/A	SR-x16	SR-x16	2T	0000_0035h	2022_3323h
1333	N/A	SR-x8	SR-x8	2T	0000_0035h	2022_3323h
1333	N/A	DR-x8	DR-x8	2T	0000_0035h	2022_3323h
1333	N/A	SR-x16	SR-x8	2T	0000_0035h	2022_3323h
1333	N/A	SR-x8	SR-x16	2T	0000_0035h	2022_3323h
1333	N/A	SR-x16	DR-x8	2T	0000_0035h	2022_3323h
1333	N/A	DR-x8	SR-x16	2T	0000_0035h	2022_3323h
1333	N/A	SR-x8	DR-x8	2T	0000_0035h	2022_3323h
1333	N/A	DR-x8	SR-x8	2T	0000_0035h	2022_3323h

1. See 2.8.9.1 [DDR3 Supply Voltage Configuration].
2. This column is used for either DIMM1 in 2 DIMM slots per channel platforms, or for DIMM2 in 3 DIMM slots per channel platforms, but not both.
3. See Table 42.

**Table 57: G34 DDR3 UDIMM Address Timings and Drive Strengths**

DDR Rate 1.5V <sup>1</sup>	DDR Rate 1.35V <sup>1</sup>	DIMM0	DIMM1 or DIMM2 <sup>2</sup>	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>3</sup>
800	800	-	SR-x16	1T	0000_0000h	2011_3222h
800	800	-	SR-x8	1T	0000_0000h	2011_3222h
800	800	-	DR-x8	1T	003B_0000h	2011_3222h
800	800	SR-x16	SR-x16	1T	0039_0039h	2022_3323h
800	800	SR-x8	SR-x8	1T	0039_0039h	2022_3323h
800	800	DR-x8	DR-x8	1T	0039_0039h	2022_3323h
800	800	SR-x16	SR-x8	1T	0039_0039h	2022_3323h
800	800	SR-x8	SR-x16	1T	0039_0039h	2022_3323h
800	800	SR-x16	DR-x8	1T	0039_0039h	2022_3323h

**Table 57: G34 DDR3 UDIMM Address Timings and Drive Strengths**

DDR Rate 1.5V <sup>1</sup>	DDR Rate 1.35V <sup>1</sup>	DIMM0	DIMM1 or DIMM2 <sup>2</sup>	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>3</sup>
800	800	DR-x8	SR-x16	1T	0039_0039h	2022_3323h
800	800	SR-x8	DR-x8	1T	0039_0039h	2022_3323h
800	800	DR-x8	SR-x8	1T	0039_0039h	2022_3323h
1066	1066	-	SR-x16	1T	0000_0000h	2011_3222h
1066	1066	-	SR-x8	1T	0000_0000h	2011_3222h
1066	1066	-	DR-x8	1T	0038_0000h	2011_3222h
1066	1066	SR-x16	SR-x16	1T	0035_0037h	2022_3323h
1066	1066	SR-x8	SR-x8	1T	0035_0037h	2022_3323h
1066	1066	DR-x8	DR-x8	1T	0035_0037h	2022_3323h
1066	1066	SR-x16	SR-x8	1T	0035_0037h	2022_3323h
1066	1066	SR-x8	SR-x16	1T	0035_0037h	2022_3323h
1066	1066	SR-x16	DR-x8	1T	0035_0037h	2022_3323h
1066	1066	DR-x8	SR-x16	1T	0035_0037h	2022_3323h
1066	1066	SR-x8	DR-x8	1T	0035_0037h	2022_3323h
1066	1066	DR-x8	SR-x8	1T	0035_0037h	2022_3323h
1333	N/A	-	SR-x16	1T	0000_0000h	2011_3222h
1333	N/A	-	SR-x8	1T	0000_0000h	2011_3222h
1333	N/A	-	DR-x8	1T	0036_0000h	2011_3222h
1333	N/A	SR-x16	SR-x16	2T	0000_0035h	2022_3323h
1333	N/A	SR-x8	SR-x8	2T	0000_0035h	2022_3323h
1333	N/A	DR-x8	DR-x8	2T	0000_0035h	2022_3323h
1333	N/A	SR-x16	SR-x8	2T	0000_0035h	2022_3323h
1333	N/A	SR-x8	SR-x16	2T	0000_0035h	2022_3323h
1333	N/A	SR-x16	DR-x8	2T	0000_0035h	2022_3323h
1333	N/A	DR-x8	SR-x16	2T	0000_0035h	2022_3323h
1333	N/A	SR-x8	DR-x8	2T	0000_0035h	2022_3323h
1333	N/A	DR-x8	SR-x8	2T	0000_0035h	2022_3323h

1. See 2.8.9.1 [DDR3 Supply Voltage Configuration].

2. This column is used for either DIMM1 in 2 DIMM slots per channel platforms, or for DIMM2 in 3 DIMM slots per channel platforms, but not both.

3. See Table 42.

**Table 58: C32 DDR3 UDIMM Address Timings and Drive Strengths**

DDR Rate 1.5V <sup>1</sup>	DDR Rate 1.35V <sup>1</sup>	DIMM0	DIMM1 or DIMM2 <sup>2</sup>	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>3</sup>
800	800	-	SR-x16	1T	0000_0000h	2011_3222h
800	800	-	SR-x8	1T	0000_0000h	2011_3222h
800	800	-	DR-x8	1T	003B_0000h	2011_3222h
800	800	SR-x16	SR-x16	1T	0039_0039h	2022_3323h
800	800	SR-x8	SR-x8	1T	0039_0039h	2022_3323h
800	800	DR-x8	DR-x8	1T	0039_0039h	2022_3323h
800	800	SR-x16	SR-x8	1T	0039_0039h	2022_3323h
800	800	SR-x8	SR-x16	1T	0039_0039h	2022_3323h
800	800	SR-x16	DR-x8	1T	0039_0039h	2022_3323h
800	800	DR-x8	SR-x16	1T	0039_0039h	2022_3323h
800	800	SR-x8	DR-x8	1T	0039_0039h	2022_3323h
800	800	DR-x8	SR-x8	1T	0039_0039h	2022_3323h
1066	1066	-	SR-x16	1T	0000_0000h	2011_3222h
1066	1066	-	SR-x8	1T	0000_0000h	2011_3222h
1066	1066	-	DR-x8	1T	0038_0000h	2011_3222h
1066	1066	SR-x16	SR-x16	1T	0035_0037h	2022_3323h
1066	1066	SR-x8	SR-x8	1T	0035_0037h	2022_3323h
1066	1066	DR-x8	DR-x8	1T	0035_0037h	2022_3323h
1066	1066	SR-x16	SR-x8	1T	0035_0037h	2022_3323h
1066	1066	SR-x8	SR-x16	1T	0035_0037h	2022_3323h
1066	1066	SR-x16	DR-x8	1T	0035_0037h	2022_3323h
1066	1066	DR-x8	SR-x16	1T	0035_0037h	2022_3323h
1066	1066	SR-x8	DR-x8	1T	0035_0037h	2022_3323h
1066	1066	DR-x8	SR-x8	1T	0035_0037h	2022_3323h
1333	1333	-	SR-x16	1T	0000_0000h	2011_3222h
1333	1333	-	SR-x8	1T	0000_0000h	2011_3222h
1333	1333	-	DR-x8	1T	0036_0000h	2011_3222h
1333	1333	SR-x16	SR-x16	2T	0000_0035h	2022_3323h
1333	1333	SR-x8	SR-x8	2T	0000_0035h	2022_3323h
1333	N/A	DR-x8	DR-x8	2T	0000_0035h	2022_3323h
1333	1333	SR-x16	SR-x8	2T	0000_0035h	2022_3323h
1333	1333	SR-x8	SR-x16	2T	0000_0035h	2022_3323h
1333	N/A	SR-x16	DR-x8	2T	0000_0035h	2022_3323h
1333	N/A	DR-x8	SR-x16	2T	0000_0035h	2022_3323h

**Table 58: C32 DDR3 UDIMM Address Timings and Drive Strengths**

DDR Rate 1.5V <sup>1</sup>	DDR Rate 1.35V <sup>1</sup>	DIMM0	DIMM1 or DIMM2 <sup>2</sup>	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>3</sup>
1333	N/A	SR-x8	DR-x8	2T	0000_0035h	2022_3323h
1333	N/A	DR-x8	SR-x8	2T	0000_0035h	2022_3323h

- See 2.8.9.1 [DDR3 Supply Voltage Configuration].
- This column is used for either DIMM1 in 2 DIMM slots per channel platforms, or for DIMM2 in 3 DIMM slots per channel platforms, but not both.
- See Table 42.

**Table 59: DDR3 SO-DIMM Address Timings and Drive Strengths for 1 DIMM per channel**

DDR Rate	DIMM0 <sup>1</sup>	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>2</sup>
800	any	1T	0000_0000h	X011_3222h
1066	any	1T	0000_0000h	X011_3222h
1333	any	1T	0000_0000h	X011_3222h

- any=SRx8, SRx16, DRx8, or DRx16.
- See Table 40.

**Table 60: DDR3 SO-DIMM Address Timings and Drive Strengths for 2 DIMMs per channel**

DDR Rate		DIMM Population <sup>2</sup>		Timing/Termination <sup>3</sup>		
1.5V <sup>1</sup>	1.35V <sup>1</sup>	DIMM0	DIMM1	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>4</sup>
800	800	-	any	1T	0000_0000h	X011_3222h
800	N/A	any	any	2T	0000_0039h	X022_3323h
1066	1066	-	any	1T	0000_0000h	X011_3222h
1066	N/A	any	any	2T	0000_0037h	X022_3323h

- See 2.8.9.1 [DDR3 Supply Voltage Configuration].
- any=SRx8 or SRx16 or DRx8 or DRx16.
- See F2x[1, 0]94[SlowAccessMode] for timing mode.
- See Table 41.



**Table 61: DDR3 RDIMM Address Timings and Drive Strengths for 1 DIMM per channel**

DDR Rate		DIMM Population	Timing/Termination <sup>2</sup>		
1.5V <sup>1</sup>	1.35V <sup>1</sup>	DIMM0	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>3</sup>	RC2/RC8 <sup>4</sup>
800	800	SR or DR	0000_0000h	X011_3222h	0h/0h
800	800	QR	0000_0000h	X022_3222h	0h/0h or 4h/0h
1066	1066	SR or DR	003C_3C3Ch	X011_3222h	0h/0h
1066	1066	QR	003C_3C3Ch	X022_3222h	0h/0h or 4h/0h
1333	1333	SR or DR	003A_3A3Ah	X011_3222h	0h/0h
1333	1333	QR	003A_3A3Ah	X022_3222h	0h/0h or 4h/0h

1. See 2.8.9.1 [DDR3 Supply Voltage Configuration].  
2. For all cases, program F2x[1, 0]94[SlowAccessMode] = 0.  
3. See Table 43 and Table 44 for additional termination values.  
4. See 2.8.9.6.2.1 [Software Control Word Initialization] for more information. For each DIMM, if the DIMM has one register then program the first RC2/RC8 data value pair given, else program the second data value pair given.

**Table 62: G34 DDR3 RDIMM Address Timings and Drive Strengths for 2 DIMMs per channel**

DDR Rate		DIMM Population <sup>2</sup>		Timing/Termination <sup>3</sup>		
1.5V <sup>1</sup>	1.35V <sup>1</sup>	DIMM0	DIMM1	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>4</sup>	RC2/RC8 <sup>5</sup>
800	800	-	SR or DR	0000_0000h	X011_3222h	0h/0h
800	800	-	QR	0000_0000h	X022_3222h	0h/0h or 4h/0h
800	800	any	any	0000_0000h	X022_3222h	4h/0h or 0h/4h
1066	1066	-	SR or DR	003C_3C3Ch	X011_3222h	0h/0h
1066	1066	-	QR	003C_3C3Ch	X022_3222h	0h/0h or 4h/0h
1066	1066	SR or DR	SR or DR	003A_3C3Ah	X022_3222h	4h/0h or 0h/4h
1066	N/A	QR	any	003A_3C3Ah	X022_3222h	4h/0h or 0h/4h
1066	N/A	any	QR	003A_3C3Ah	X022_3222h	4h/0h or 0h/4h
1333	1333	-	SR or DR	003A_3A3Ah	X011_3222h	0h/0h
1333	N/A	-	QR	003A_3A3Ah	X022_3222h	0h/0h or 4h/0h
1333	1333	SR or DR	SR or DR	0038_3A38h	X022_3222h	4h/0h

- See 2.8.9.1 [DDR3 Supply Voltage Configuration].
- any = SR, DR, or QR
- For all cases, program F2x[1, 0]94[SlowAccessMode] = 0.
- See Table 43 and Table 44 for additional termination values.
- See 2.8.9.6.2.1 [Software Control Word Initialization] for more information.  
For each DIMM, if the DIMM has one register then program the first RC2/RC8 data value pair given, else program the second data value pair given.

**Table 63: C32 DDR3 RDIMM Address Timings and Drive Strengths for 2 DIMMs per channel**

DDR Rate		DIMM Population <sup>2</sup>		Timing/Termination <sup>3</sup>		
1.5V <sup>1</sup>	1.35V <sup>1</sup>	DIMM0	DIMM1	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>4</sup>	RC2/RC8 <sup>5</sup>
667	667	any	any	0000_0000h	X022_3222h	4h/0h or 0h/4h
800	800	-	SR or DR	0000_0000h	X011_3222h	0h/0h
800	800	-	QR	0000_0000h	X022_3222h	0h/0h or 4h/0h
800	800	SR or DR	SR or DR	0000_0000h	X022_3222h	4h/0h or 0h/4h
800	N/A	any	QR	0000_0000h	X022_3222h	4h/0h or 0h/4h
800	N/A	QR	any	0000_0000h	X022_3222h	4h/0h or 0h/4h
1066	1066	-	SR or DR	003C_3C3Ch	X011_3222h	0h/0h

**Table 63: C32 DDR3 RDIMM Address Timings and Drive Strengths for 2 DIMMs per channel**

DDR Rate		DIMM Population <sup>2</sup>		Timing/Termination <sup>3</sup>		
1.5V <sup>1</sup>	1.35V <sup>1</sup>	DIMM0	DIMM1	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>4</sup>	RC2/RC8 <sup>5</sup>
1066	1066	-	QR	003C_3C3Ch	X022_3222h	0h/0h or 4h/0h
1066	1066	SR or DR	SR or DR	003A_3C3Ah	X022_3222h	4h/0h or 0h/4h
1333	1333	-	SR or DR	003A_3A3Ah	X011_3222h	0h/0h
1333	N/A	-	QR	003A_3A3Ah	X022_3222h	0h/0h or 4h/0h
1333	1333	SR	SR	0038_3A38h	X022_3222h	4h/0h

1. See 2.8.9.1 [DDR3 Supply Voltage Configuration].
2. any = SR, DR, or QR.
3. For all cases, program F2x[1, 0]94[SlowAccessMode] = 0.
4. See Table 43 and Table 44 for additional termination values.
5. See 2.8.9.6.2.1 [Software Control Word Initialization] for more information.  
For each DIMM, if the DIMM has one register then program the first RC2/RC8 data value pair given, else program the second data value pair given.

**Table 64: G34 DDR3 RDIMM Address Timings and Drive Strengths for 3 DIMMs per channel**

DDR Rate		DIMM Population <sup>2</sup>			Timing/Termination <sup>3</sup>		
1.5V <sup>1</sup>	1.35V <sup>1</sup>	DIMM0	DIMM1	DIMM2	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>4</sup>	RC2/ RC8 <sup>5</sup>
667	667	-	-	SR or DR	0000_0000h	X011_3222h	0h/0h
667	667	SR or DR	-	SR or DR	0000_0000h	X022_3222h	4h/0h
667	667	SR or DR	any	SR or DR	0038_0038h	X011_3222h	4h/0h or 0h/4h
800	800	-	-	SR or DR	0000_0000h	X011_3222h	0h/0h
800	800	SR or DR	-	SR or DR	0000_0000h	X022_3222h	4h/0h
800	800	SR or DR	SR or DR	SR or DR	0038_0038h	X011_3222h	4h/0h or 0h/4h
800	N/A	SR or DR	QR	SR or DR	0038_0038h	X011_3222h	4h/0h or 0h/4h
800	800	-	QR	-	0000_0000h	X022_3222h	0h/0h or 4h/0h
800	800	-	QR	SR or DR	0000_0000h	X022_3222h	4h/0h or 0h/4h
1066	1066	-	-	SR or DR	003C_3C3Ch	X011_3222h	0h/0h
1066	1066	SR or DR	-	SR or DR	003A_3C3Ah	X022_3222h	4h/0h
1066	1066	SR	SR	SR	0037_3C37h	X011_3222h	4h/0h
1066	N/A	SR or DR	SR or DR	DR	0037_3C37h	X011_3222h	4h/0h
		SR or DR	DR	SR or DR	0037_3C37h	X011_3222h	4h/0h
		DR	SR or DR	SR or DR	0037_3C37h	X011_3222h	4h/0h
1066	1066	-	QR	-	003C_3C3Ch	X022_3222h	0h/0h or 4h/0h
1333	1333	-	-	SR or DR	003A_3A3Ah	X011_3222h	0h/0h
1333	1333	SR or DR	-	SR or DR	0038_3A38h	X022_3222h	4h/0h
1333	N/A	SR	SR	SR	0034_3A34h	X011_3222h	4h/0h or 0h/4h

1. See 2.8.9.1 [DDR3 Supply Voltage Configuration].

2. any = SR, DR, or QR

3. For all cases, program F2x[1, 0]94[SlowAccessMode] = 0.

4. See Table 42 and Table 43 for additional termination values.

5. See 2.8.9.6.2.1 [Software Control Word Initialization] for more information.

For each DIMM, if the DIMM has one register, program the first RC2/RC8 data value pair given, else program the second data value pair given.

**Table 65: C32 DDR3 RDIMM Address Timings and Drive Strengths for 3 DIMMs per channel**

DDR Rate		DIMM Population <sup>2</sup>			Timing/Termination <sup>3</sup>		
1.5V <sup>1</sup>	1.35V <sup>1</sup>	DIMM0	DIMM1	DIMM2	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00 <sup>4</sup>	RC2/ RC8 <sup>5</sup>
667	667	-	-	SR or DR	0000_0000h	X011_3222h	0h/0h
667	667	SR or DR	-	SR or DR	0000_0000h	X022_3222h	4h/0h
667	667		QR	SR or DR	0000_0000h	X022_3222h	4h/0h or 0h/4h
667	667	SR or DR	SR or DR	SR or DR	0038_0038h	X022_3222h	4h/0h or 0h/4h
667	667	SR or DR	QR	SR or DR	0038_0038h	X022_3222h	4h/0h or 0h/4h
800	800	-	-	SR or DR	0000_0000h	X011_3222h	0h/0h
800	800	SR or DR	-	SR or DR	0000_0000h	X022_3222h	4h/0h
800	N/A		QR	SR or DR	0000_0000h	X022_3222h	4h/0h or 0h/4h
800	800	SR or DR	SR or DR	SR or DR	0038_0038h	X011_3222h	4h/0h or 0h/4h
800	800	-	QR	-	0000_0000h	X022_3222h	0h/0h or 4h/0h
1066	1066	-	-	SR or DR	003C_3C3Ch	X011_3222h	0h/0h
1066	1066	SR or DR	-	SR or DR	003A_3C3Ah	X022_3222h	4h/0h
1066	N/A	SR	SR	SR	0037_3C37h	X011_3222h	4h/0h
1333	1333	-	-	SR or DR	003A_3A3Ah	X011_3222h	0h/0h
1333	1333	SR	-	SR	0038_3A38h	X022_3222h	4h/0h

1. See 2.8.9.1 [DDR3 Supply Voltage Configuration].  
2. any = SR, DR, or QR.  
3. For all cases, program F2x[1, 0]94[SlowAccessMode] = 0.  
4. See Table 43 and Table 44 for additional termination values.  
5. See 2.8.9.6.2.1 [Software Control Word Initialization] for more information.  
For each DIMM, if the DIMM has one register, program the first RC2/RC8 data value pair given, else program the second data value pair given.

### 2.8.9.6 DRAM Device and Controller Initialization

BIOS initializes the DRAM devices and the controller using either a hardware or software controlled sequence. Registered DDR3 DIMMs must be initialized under software control. Unbuffered DDR3 DIMMs may be initialized using either the hardware or software sequence.

The hardware controlled sequence is as follows:

- Program F2x[1, 0]90[InitDram] = 1. See Note below.

The software controlled sequences are listed in subsections as follows:

- For DDR2 DIMMs, perform the software device initialization as described in 2.8.9.6.1. See Note below.

- For DDR3 DIMMs, perform the software device initialization as described in 2.8.9.6.2. See Note below.

Note: BIOS must observe additional requirements for changing the PLL frequency when setting F2x[1, 0]90[InitDram] or F2x[1, 0]7C[EnDramInit]. See 2.8.9.8 [DRAM Channel Frequency Change] for more information.

DRAM initialization completes after the hardware-controlled initialization process completes or when the BIOS-controlled initialization process completes (F2x[1, 0]7C[EnDramInit] is written from 1 to 0).

For DDR2, part of the initialization sequence includes writing mode register set (MRS) values to the DDR2 DRAM. The values written to MRS and EMRS in DRAM devices are determined as follows when using the hardware-controlled sequence:

- MRS[2:0] burst length (BL): based on F2x[1, 0]90[Width128 and BurstLength32].
- MRS[3] burst type (BT): interleave.
- MRS[6:4] CAS latency: based on F2x[1, 0]88[Tcl].
- MRS[7] test mode (TM): normal mode.
- MRS[8] DLL reset (DLL): controlled as required by the initialization sequence.
- MRS[11:9] write recovery for auto pre-charge (WR): based on F2x[1, 0]84[Twr] or F2x[1, 0]88[Twr] depending on memory speed.
- MRS[12] active power down exit time (PD): fast exit (although the mode is not supported).
- EMRS(1)[0]: DLL enable (DLL): enabled.
- EMRS(1)[1]: output driver impedance control (DIC): based on F2x[1, 0]90[DramDrvWeak].
- EMRS(1)[6,2]: Rtt: based on F2x[1, 0]90[DramTerm].
- EMRS(1)[5:3]: additive latency: fixed at 0.
- EMRS(1)[9:7]: OCD calibration program: controlled as required by the initialization sequence (but not calibrated).
- EMRS(1)[10]: DQS bar: based on F2x[1, 0]90[DisDqsBar].
- EMRS(1)[11]: RDQS: based on F2x[1, 0]94[RDqsEn].
- EMRS(1)[12]: Qoff: output buffer enabled.
- EMRS(2)[7]: SRF: high temperature self refresh rate enable, based on F2x[1, 0]90[SelfRefRateEn].

For DDR3 unbuffered DIMMS, a similar initialization sequence is invoked; DDR3 registered DIMMs do not support hardware-controlled initialization. The values written to the DRAM device's MRs when using the hardware-controlled sequence are determined as follows:

- MR0[1:0] burst length and control method (BL): based on F2x[1, 0]84[BurstCtrl].
- MR0[3] burst type (BT): interleaved.
- MR0[6:4,2] read CAS latency (CL): based on F2x[1, 0]88[Tcl].
- MR0[7] test mode (TM): normal mode.
- MR0[8] DLL reset (DLL Reset): controlled as required by the initialization sequence.
- MR0[11:9] write recovery for auto-precharge (WR): based on F2x[1, 0]84[Twr].
- MR0[12] precharge power-down mode select (PPD): based on F2x[1, 0]84[PchgPDMoSel].
- MR1[0] DLL disable (DLL Dis): DLL enabled.
- MR1[1] output driver impedance control (DIC): normal.
- MR1[6,2] nominal termination resistance of ODT (RTT): based on F2x[1, 0]90[DramTerm].
- MR1[4:3] additive latency (AL): AL is disabled.
- MR1[7] write leveling enable (Level): controlled as required by the initialization sequence.
- MR1[11]: TDQS: based on F2x[1, 0]94[RDqsEn].
- MR1[12] output disable (QOFF): based on F2x[1, 0]84[Qoff].
- MR2[2:0] partial array self refresh (PASR): full array.

- MR2[5:3] CAS write latency (CWL): based on F2x[1, 0]84[Tcwl].
- MR2[6] auto self refresh method (ASR): based on F2x[1, 0]84[ASR].
- MR2[7] self refresh temperature range (SRT): based on F2x[1, 0]84[ASR and SRT].
- MR3[1:0] multi purpose register address location (MPR Location): based on F2x[1, 0]84[MprLoc].
- MR3[2] multi purpose register (MPR): based on F2x[1, 0]84[MprEn].

The processor does not support the use of speculative system-memory reads and writes to determine the size of system memory. It is expected that BIOS determines the size of system memory by reading DIMM SPD information or an equivalent means.

### 2.8.9.6.1 Software DDR2 Device Initialization

The following BIOS controlled software initialization procedure applies to each DRAM controller and properly initializes all the DDR2 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. Configure the DCT registers, including MemClkFreq and MemClkFreqVal.
2. Program F2x[1, 0]7C[EnDramInit] = 1. See Note.
3. Wait 200 us.
4. Program F2x[1, 0]7C[DeassertMemRstX] = 1.
5. Wait 200 us.
6. Program F2x[1, 0]7C[AssertCke] = 1.
7. Wait 400 ns.
8. Send Precharge All command.
9. Send EMRS(2).
10. Send EMRS(3).
11. Send EMRS(1) with MrsAddress[6,2] = 00b at this time.
12. Send MRS with MrsAddress[8] = 1.
13. Wait 200 MEMCLKs.
14. Send Precharge All command.
15. Send two Auto Refresh commands.
16. Send MRS with MrsAddress[8] = 0.
17. Send EMRS(1) with MrsAddress[9:7] = 111b and set MrsAddress[6,2]=00b at this time.
18. Send EMRS(1) with MrsAddress[9:7] = 000b.
19. Program F2x[1, 0]7C[EnDramInit] = 0.

Note: BIOS must observe additional requirements for changing the PLL frequency when setting F2x[1, 0]7C[EnDramInit]. See 2.8.9.8 [DRAM Channel Frequency Change] for more information.

Send Precharge All command is accomplished as follows:

1. Program F2x[1, 0]7C[SendPchgAll] = 1.
2. Wait Trp.

Send Auto Refresh command is accomplished as follows:

1. Program F2x[1, 0]7C[SendAutoRefresh] = 1.
2. Wait for F2x[1, 0]7C[SendAutoRefresh] = 0.

Send MRS command is accomplished by programming the [The DRAM Initialization Register] F2x[1, 0]7C register as follows:

1. Program MrsBank = 000b.
2. If EnDramInit=0 program MrsChipSel=*chipselect*; otherwise all chip selects are automatically selected.



3. Program MrsAddress[2:0] = burst length (BL): based on F2x[1, 0]90[Width128 and BurstLength32].
  - 010b = 4-beat burst length.
  - 011b = 8-beat burst length.
4. Program MrsAddress[3] = 1.
5. Program MrsAddress[6:4] = CAS latency based on the F2x[1, 0]88[Tcl] field.
6. Program MrsAddress[8] = DLL reset (DLL), controlled as required by the initialization sequence.
7. Program MrsAddress[11:9] = write recovery for auto pre-charge (WR): based on F2x[1, 0]84[Twr] or F2x[1, 0]88[Twr] depending on memory speed.
8. Set all other bits in MrsAddress to zero.
9. Set SendMrsCmd = 1.
10. Wait for SendMrsCmd = 0.

Send EMRS(1) command is accomplished by programming F2x[1, 0]7C as follows:

1. Program MrsBank = 001b.
2. If EnDramInit=0 program MrsChipSel=*target chip select*; otherwise all chip selects are automatically selected.
3. Program MrsAddress[0] = 0.
4. Program MrsAddress[1] = output driver impedance control (DIC): based on F2x[1, 0]90[DramDrvWeak].
5. Program MrsAddress[6,2] = Rtt: based on F2x[1, 0]90[DramTerm].
6. Program MrsAddress[9:7] = OCD calibration program: controlled as required by the initialization sequence (but not calibrated).
7. Program MrsAddress[10] = DQS bar based on F2x[1, 0]90[DisDqsBar].
8. Program MrsAddress[11] = RDQS based on F2x[1, 0]94[RDqsEn] for unbuffered DIMMs. Program MrsAddress[11] = 0 for registered DIMMs with x4 devices or with x8 devices when only x8 devices are present on the channel, and MrsAddress[11] = 1 for registered DIMMs with x8 devices when both x4 and x8 devices are present on the channel.
9. Set all other bits in MrsAddress to zero.
10. Set SendMrsCmd = 1.
11. Wait for SendMrsCmd = 0.

Send EMRS(2) command is accomplished by programming F2x[1, 0]7C as follows:

1. Program MrsBank = 010b.
2. If EnDramInit=0 program MrsChipSel=*target chip select*; otherwise all chip selects are automatically selected.
3. Program MrsAddress[7] = SRF: high temperature self refresh rate enable, based on F2x[1, 0]90[SelfRefrateEn].
4. Set all other bits in MrsAddress to zero.
5. Set SendMrsCmd = 1.
6. Wait for SendMrsCmd = 0.

Send EMRS(3) command is accomplished by programming F2x[1, 0]7C as follows:

1. Program MrsBank = 011b.
2. If EnDramInit=0 program MrsChipSel=*target chip select*; otherwise all chip selects are automatically selected.
3. Program MrsAddress[15:0] = 0.
4. Set SendMrsCmd = 1.
5. Wait for SendMrsCmd = 0.

### 2.8.9.6.2 Software DDR3 Device Initialization

The following BIOS controlled software initialization procedure applies to each DRAM controller to properly

initialize all 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). This procedure is required to support registered DDR3 DIMMs. This procedure may also be used to support unbuffered DDR3 DIMMs:

1. Configure the DCT registers, including MemClkFreq and MemClkFreqVal.
2. Program `F2x[1, 0]7C[EnDramInit] = 1`. Note: BIOS must observe additional requirements for changing the PLL frequency when setting `F2x[1, 0]7C[EnDramInit]`. See 2.8.9.8 [DRAM Channel Frequency Change] for more information.
3. Wait 200 us.
4. Program `F2x[1, 0]7C[DeassertMemRstX] = 1`.
5. Wait 500 us.
6. Program `F2x[1, 0]7C[AssertCke] = 1`.
7. Wait 360 ns.

The following steps are performed with registered DIMMs only and must be done for each chip select pair:

8. Send RCW(0), RCW(1), and RCW(2). See 2.8.9.6.2.1 for details.
9. Wait 6 us.
10. Send RCW(3), RCW(4), and RCW(5).
11. Send RCW(6) and RCW(7) for custom settings at this time, as directed by the DIMM manufacturer's data sheet.

The following steps are performed once for each channel with unbuffered DIMMs and once for each chip select with registered DIMMs:

12. Send EMRS(2). See Note below.
13. Send EMRS(3). Ordinarily at this time, `MrsAddress[2:0] = 000b`.
14. Send EMRS(1).
15. Send MRS with `MrsAddress[8] = 1`.

Note: Unbuffered DIMMs optionally have address bits rearranged from the edge connector to the second rank of a dual rank DIMM. This feature is called address mirroring. The BIOS must program `F2x[1, 0][5C:40][OnDimmMirror] = 1` prior to sending the MR commands used for device initialization if SPD byte 63 indicates that address mapping is mirrored.

The following steps are performed for all DIMM types:

16. Send two ZQCL commands.
17. Program `F2x[1, 0]7C[EnDramInit] = 0`.

Send ZQCL command is accomplished by programming `F2x[1, 0]7C` as follows:

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

Send MRS command for DDR3 initialization is accomplished by programming `F2x[1, 0]7C` as follows:

1. Program `MrsBank = 000b`.
2. If `EnDramInit=0` program `MrsChipSel=target chip select`; otherwise all chip selects are automatically selected.
3. Program `MrsAddress[1:0] = burst length and control method (BL): based on F2x[1, 0]84[BurstCtrl]`.
4. Program `MrsAddress[3] = 1`.
5. Program `MrsAddress[6:4,2] = read CAS latency (CL): based on F2x[1, 0]88[Tcl]`.  
0000b = 4 MEMCLKs.

0010b = 5 MEMCLKs.  
 0100b = 6 MEMCLKs.  
 0110b = 7 MEMCLKs.  
 1000b = 8 MEMCLKs.  
 1010b = 9 MEMCLKs.  
 1100b = 10 MEMCLKs.  
 1110b = 11 MEMCLKs.  
 0001b = 12 MEMCLKs.

6. Program MrsAddress[11:9] = write recovery for auto-precharge (WR): based on F2x[1, 0]84[Twr].
7. Program MrsAddress[12] = precharge powerdown mode select (PPD): based on F2x[1, 0]84[PchgPDMoSel].
8. Set all other bits in MrsAddress to zero.
9. Set SendMrsCmd = 1.
10. Wait for SendMrsCmd = 0.

Send EMRS(1) command for DDR3 initialization is accomplished by programming F2x[1, 0]7C as follows:

1. Program MrsBank = 001b.
2. If EnDramInit=0 program MrsChipSel=*target chip select*; otherwise all chip selects are automatically selected.
3. Program MrsAddress[5,1] = output driver impedance control (DIC): based on F2x[1, 0]84[DrvImpCtrl].
4. Program MrsAddress[9,6,2] = nominal termination resistance of ODT (RTT): based on F2x[1, 0]84[DramTerm].
5. Program MrsAddress[11] = TDQS based on F2x[1, 0]94[RDqsEn] for unbuffered DIMMs. Program MrsAddress[11] = 0 for registered DIMMs with x4 devices or with x8 devices when only x8 devices are present on the channel, and MrsAddress[11] = 1 for registered DIMMs with x8 devices when both x4 and x8 devices are present on the channel.
6. Program MrsAddress[12] = output disable (QOFF): based on F2x[1, 0]84[Qoff].
7. Set all other bits in MrsAddress to zero.
8. Set SendMrsCmd = 1.
9. Wait for SendMrsCmd = 0.

Send EMRS(2) command for DDR3 initialization is accomplished by programming the F2x[1, 0]7C as follows:

1. Program MrsBank = 010b.
2. If EnDramInit=0 program MrsChipSel=*target chip select*; otherwise all chip selects are automatically selected.
3. Program MrsAddress[5:3] = CAS write latency (CWL): based on F2x[1, 0]84[Tcwl].
4. Program MrsAddress[6] = auto self refresh method (ASR): based on F2x[1, 0]84[ASR].
5. Program MrsAddress[7] = self refresh temperature range (SRT): based on F2x[1, 0]84[ASR and SRT].
6. Program MrsAddress[10:9] = dynamic termination during writes (RTT\_WR): based on F2x[1, 0]84[DramTermDyn].
7. Set all other bits in MrsAddress to zero.
8. Set SendMrsCmd = 1.
9. Wait for SendMrsCmd = 0.

Send EMRS(3) command for DDR3 initialization is accomplished by programming F2x[1, 0]7C as follows:

1. Program MrsBank = 011b.
2. If EnDramInit=0 program MrsChipSel=*target chip select*; otherwise all chip selects are automatically selected.
3. Program MrsAddress[1:0] = multi purpose register address location (MPR Location): based on F2x[1, 0]84[MprLoc].

4. Program MrsAddress[2] = multi purpose register (MPR): based on F2x[1, 0]84[MprEn].
5. Set all other bits in MrsAddress to zero.
6. Set SendMrsCmd = 1.
7. Wait for SendMrsCmd = 0.

#### 2.8.9.6.2.1 Software Control Word Initialization

DDR3 register devices on RDIMMs contain up to 16 control words, referred to as RC0 to RC15. Each control word is four bits. These devices are programmed at the bus using an RCW command by (a) presenting the 4-bit address of the control word on [BA2, A2, A1, A0], (b) presenting the 4-bit write data on [BA1, BA0, A4, A3], and (c) asserting both chip selects of a chip select pair.

An RCW(n) command for DDR3 register device initialization is accomplished by programming F2x[1, 0]7C and F2x[1, 0]A8[CtrlWordCS] as follows:

1. Program MrsBank and MrsAddress.
  - n = [BA2, A2, A1, A0].
  - data = [BA1, BA0, A4, A3].
  - Set all other bits in MrsAddress to zero.
2. Program F2x[1, 0]A8[CtrlWordCS]=bit mask for the target chip selects.
3. Set SendControlWord = 1.
4. Wait for SendControlWord = 0.

Based on the number of ranks on the DIMM and the total bus load, the BIOS programs RC0 to RC15 for the DIMM according to the information in [Table 66](#):

**Table 66. DDR3 RDIMM Register Control Word Values**

Control Word	Control Data
RC0	2h
RC1	For a 1 rank DIMM data=0Ch, else data=0h
RC2	Based on bus load. See <a href="#">Table 62</a> , and <a href="#">Table 64</a> for data value.
RC3	data = SPD byte 70 bits [7:4]
RC4	data = SPD byte 71 bits [3:0]
RC5	data = SPD byte 71 bits [7:4]
RC8	Based on bus load. See <a href="#">Table 62</a> , and <a href="#">Table 64</a> for data value.
RC9	0Dh
RC10	Based on MEMCLK frequency: 0h; (frequency <= 400 MHz) 1h; (400 MHz < frequency <= 533 MHz) 2h; (533 MHz < frequency <= 667 MHz)
RC11	Based on Register VDD Operating Voltage: 0h: 1.5V 1h: 1.35V See <a href="#">2.8.9.1 [DDR3 Supply Voltage Configuration]</a> .
RC12 - RC15	0h

**Table 66. DDR3 RDIMM Register Control Word Values**

Control Word	Control Data
Notes: 1. RC6 and RC7 are reserved. 2. SPD byte 63 bits [1:0] indicates the number of register devices on the quad rank DIMM. For two devices, BIOS must initialize register control words of both devices on the DIMM by repeating the sequence for each chip select pair.	

### 2.8.9.7 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 DDR frequency change (see 2.8.9.8 [DRAM Channel Frequency Change]) and before any memory access (including MRS commands), BIOS must perform phy fence training for each channel using the following steps:

1. Set  $F2x[1, 0]9C\_x08[PhyFenceTrEn]=1$ .
2. Wait 200 MEMCLKs.
3. Clear  $F2x[1, 0]9C\_x08[PhyFenceTrEn]=0$ .
4. BIOS reads the phase recovery engine registers  $F2x[1, 0]9C\_x[51:50]$  and  $F2x[1, 0]9C\_x52$ .
5. Calculate the average of the fine delay values of all byte lanes and subtract 8.
6. Write the value to  $F2x[1, 0]9C\_x0C[PhyFence]$ .
7. BIOS rewrites  $F2x[1, 0]9C\_x04$ , DRAM Address/Command Timing Control Register delays for both channels.

### 2.8.9.8 DRAM Channel Frequency Change

BIOS configures the channel frequency by programming the target frequency in the DCT ( $F2x[1, 0]94[MemClkFreq]$ ) and triggering the DCT to change the PLL frequency in the phy. BIOS accomplishes this during the boot process by setting one of:

- $F2x[1, 0]7C[EnDramInit]$
- $F2x[1, 0]90[InitDram]$
- $F2x[1, 0]90[ExitSelfRef]$
- $F2x[1, 0]94[MemClkFreqVal]$ . Note: If the DCT has not been initialized by one of the other three control bits then setting  $MemClkFreqVal$  will not affect a frequency change in the phy or on the bus.

BIOS observes the following requirements for changing the PLL frequency under all boot conditions (including restoring the DCT state when booting from the S3 state):

- BIOS disables the phy auto-compensation engine a minimum of 5 us prior to changing the PLL frequency by programming  $F2x[1, 0]9C\_x08[DisAutoComp] = 1$ .
- BIOS re-enables auto-compensation ( $DisAutoComp = 0$ ) after the frequency change is complete and waits 750 us before the next memory access.
- If BIOS programs  $F2x[1, 0]90[DisDllShutdownSR]$  to 0, BIOS must also:
  - Program  $F2x[1, 0]9C\_x0D0FE007[DllLockTime] = 013Dh$ .
  - Program  $F2x[1, 0]9C\_x0D0FE006[PllLockTime] = 07D0h$  prior to changing the PLL frequency.
  - Program  $F2x[1, 0]9C\_x0D0FE006[PllLockTime] = 001Ch$  after the frequency change is complete.
- BIOS observes all of the individual requirements for accessing DCT registers which may cause a frequency change in the phy ( $EnDramInit$ ,  $InitDram$ ,  $ExitSelfRef$ , and  $MemClkFreqVal$ ).

- When both DCTs are enabled in ungang mode, BIOS initializes the PLL frequency of each DCT in order by ensuring that the phy auto-compensation is disabled on both DCTs prior to any change in PLL frequency and that the frequency change has completed on both DCTs prior to re-enabling auto-compensation.
- BIOS must not change the PLL frequency after DRAM has exited from self-refresh.
- BIOS must not change the PLL frequency after DRAM device initialization for DDR2 DIMMs is complete and after DRAM training for DDR3 DIMMs is complete.

### 2.8.9.9 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 [2.8.9.6 \[DRAM Device and Controller Initialization\]](#)). It may be entirely BIOS controlled or BIOS may use hardware to assist with the training process in the case of DDR3.

If the DCTs are to be operated in ganged mode (see [2.8 \[DRAM Controllers \(DCTs\)\]](#)) then the training algorithms are done in ganged mode. Likewise, if the DCTs are ungang then the training is done ungang. However, when in ganged mode, training should use the worst case  $F2x[1, 0]78[MaxRdLatency]$  that exists between either DRAM channel.

BIOS must program  $MSRC001\_1023[WbEnhWsbDis]=1$  before training and program  $MSRC001\_1023[WbEnhWsbDis]=0$  when DRAM training is complete.

For revision DA-C and C3, if NB P-states are enabled, BIOS must program  $MSRC001\_001F[GfxNbPstateDis]=1$  before training.

DRAM training of the ECC byte lanes is accomplished after the data lanes are trained. This is described in section [2.8.9.9.4 \[ECC Byte Lane Training\]](#).

DDR3 training is a superset of DDR2 training. DDR3 requires the same training process for DQS receiver enable and DQ-DQS position that is accomplished for DDR2. See [2.8.9.9.2 \[DQS Receiver Enable Training\]](#) and [2.8.9.9.3 \[DQS Position Training\]](#). While DDR2 uses a star topology for command and address, DDR3 employs a flyby topology where each tap point on the command and address bus is high impedance. Write levelization (DDR3-defined tDQSS margining) solves the MEMCLK to DQS skew problem caused by the flyby topology by using the phy's ability to delay the launch of each DQS going to the DIMM such that at each DRAM chip, DQS is seen to coalesce with incoming MEMCLK. Levelization is done per channel and per DIMM. Levelization can be performed on each channel in parallel regardless of whether the channels are ganged or not.

Some restrictions for write levelization training follow:

- Write levelization must be done before DQS receiver enable and DQ-DQS position training.
- No memory reads or writes to DRAM should occur before write levelization training; otherwise, write levelization training may fail.

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

#### 2.8.9.9.1 Write Levelization Training

Write levelization involves using the phase recovery engine in the phy to detect the edge of DQS with respect to the memory clock on the DRAM for write accesses to each lane. Write levelization is performed only for systems with DDR3 DIMMs.



Training is accomplished on a per channel, per DIMM basis. If the target frequency is greater than the lowest supported MEMCLK frequency then BIOS performs two passes; otherwise, only one pass is required.

- Pass 1: Configure the DRAM interface and the memory subsystem for the lowest supported MEMCLK frequency (see F2x[1, 0]94[MemClkFreq]) and execute the steps below.
- Pass 2: Configure the DRAM interface and the memory subsystem for the target MEMCLK frequency (See 2.8.9.8 [DRAM Channel Frequency Change]) and execute the steps below.

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

- Ensure that F2x[1, 0]8C[DisAutoRefresh] = 1.
- Ensure that F2x[1, 0]94[ZqcsInterval] = 00b.

For each DIMM or chip select pair of a quad rank DIMM:

1. Prepare the DIMMs for write levelization using DDR3-defined MR commands. See F2x[1, 0]7C.
  - A quad rank DIMM is treated as two DIMMs. In the following steps, the target rank and next subsequent rank of a quad rank DIMM are referred to as the target DIMM. The remaining two ranks are treated as a non-target DIMM.
  - A. Configure the output driver and on-die termination of the target DIMM as follows:
    - For the first rank of the target DIMM, enable write leveling mode and enable the output driver.
    - For all other ranks of the target DIMM, enable write leveling mode and disable the output driver.
    - For two or more DIMMs per channel, program Rtt\_Nom of the target rank to the corresponding specified Rtt\_Wr termination. Otherwise, configure Rtt\_Nom of the target DIMM as normal. See 2.8.9.5.7 [DRAM ODT Control].
  - B. Configure Rtt\_Nom on the non-target DIMMs as normal. See 2.8.9.5.7.
  - C. Wait 40 MEMCLKs.
2. Configure the phy for write levelization training:
  - A. Program F2x[1, 0]9C\_x08[WrtLvTrEn]=0.
  - B. Program F2x[1, 0]9C\_x08[TrDimmSel] to specify the target DIMM to be trained.
  - C. Program F2x[1, 0]9C\_x08[WrLvOdt[3:0]] for the current memory configuration. See 2.8.9.5.7 [DRAM ODT Control].
  - D. Program F2x[1, 0]9C\_x08[WrLvOdtEn]=1.
  - E. Wait 10 MEMCLKs to allow for ODT signal settling.
  - F. For each lane program an initial value to registers F2x[1, 0]9C\_x52:F2x[1, 0]9C\_x[51:50] to set the gross and fine delay. See 2.8.9.9.1.1 [Write Leveling Seed Value].
3. Perform write leveling of the devices on the DIMM:
  - A. Program F2x[1, 0]9C\_x08[WrtLvTrEn]=1.
  - B. Wait 200 MEMCLKs.
  - C. Program F2x[1, 0]9C\_x08[WrtLvTrEn]=0.
  - D. Read from registers F2x[1, 0]9C\_x52:F2x[1, 0]9C\_x[51:50] to get the gross and fine delay settings for the target DIMM and save these values.
4. Disable ODT.
  - A. Program F2x[1, 0]9C\_x08[WrLvOdtEn]=0.
  - B. Wait 10 MEMCLKs to allow for ODT signal settling.
5. Program the target DIMM back to normal operation (see step #1 above)
  - Configure all ranks of the target DIMM for normal operation.
  - Enable the output drivers of all ranks of the target DIMM.
  - For two or more DIMMs per channel, program Rtt\_Nom of the target DIMM to the normal operating termination. See 2.8.9.5.7.
6. For each lane, calculate and program the corresponding write DQS delay values for F2x[1, 0]9C\_x[4A:30][WrDqsGrossDly, WrDqsFineDly].
  - WrDqsFineDly = PhRecFineDlyByte.



- $WrDqsGrossDly = SeedGross + PhRecGrossDlyByte - SeedPreGross$ .
- If the total delay is negative (the most significant bit of the  $WrDqsGrossDly$  field as defined by  $F2x[1, 0]9C\_x[4A:30]$  is set to one) then program the total delay for that lane to zero.

After training is complete, program  $F2x[1, 0]8C[DisAutoRefresh] = 0$  and  $F2x[1, 0]94[ZqcsInterval]$  to the proper interval for the current memory configuration.

### 2.8.9.9.1.1 Write Leveling Seed Value

The seed value for pass 1 of write levelization training is design and platform specific and should be determined by characterization for best performance. The seed value represents the total delay difference between clock delay and transmit DQS delay measured at the DRAM pins, in 1 UI/32 increments. The seed value must fall within +/- 1 UI, including PVT and jitter, of the measured delay difference.

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: Refer to [Table 67](#) to find a configuration specific seed value.
  - Pass 2:
    - If (registered DIMMs) then  $RegisterDelay = (RC2[0] ? 0x30: 0x20)$ . See [Table 66](#) for more information on  $RC2[0]$ .  
else  $RegisterDelay = 0$ .
    - $SeedTotalPreScaling = (\text{the total delay value in } F2x[1, 0]9C\_x[4A:30] \text{ from pass 1 of write levelization training}) - RegisterDelay$ .
    - $SeedTotal = SeedTotalPreScaling * \text{target frequency} / \text{lowest supported frequency} + RegisterDelay$ .
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  $F2x[1, 0]9C\_x52:F2x[1, 0]9C\_x[51:50][PhRecFineDlyByte] = SeedFine$ .
6. Program  $F2x[1, 0]9C\_x52:F2x[1, 0]9C\_x[51:50][PhRecGrossDlyByte] = SeedPreGross$ .

**Table 67. Write Leveling Seed Values**

DIMMs	Seed Value
UDIMMs	1Ah
SO-DIMMs	12h
RDIMMs at 333 MHz	If $RC2[0]=0:3Bh$ else: 4Bh
RDIMMs at 400 MHz	If $RC2[0]=0:41h$ else: 51h
Notes:	
1. UDIMM seed value assumes DDR800 speed.	
2. See <a href="#">Table 66</a> for more information on $RC2[0]$ .	

### 2.8.9.9.2 DQS Receiver Enable Training

Receiver enable delay training is used to dynamically determine the optimal delay value for [The DRAM DQS Receiver Enable Timing Control Registers] F2x[1, 0]9C\_x[2B:10]. The optimal DQS receiver enable delay value is platform and load specific, and occurs in the middle of a received read preamble.

Training is accomplished on a per channel, per DIMM, per lane basis. The lane size is 8-bits. The following describes the steps used for each pass of receiver enable training for each channel:

- Program F2x[1, 0]78[MaxRdLatency]=0C8h.
- Program F2x[1, 0]78[DqsRcvEnTrain]=1.

For each DIMM:

- A quad rank DIMM is treated as two dual rank DIMMs. BIOS executes the steps below twice.
  1. Program F2x[1, 0]9C\_x[3:0]03:F2x[1, 0]9C\_x[3:0]0[2:1] to the value in F2x[1, 0]9C\_x[4A:30] for each lane.
  2. Program F2x[1, 0]9C\_x[3:0]0[6:5] and F2x[1, 0]9C\_x[3:0]07 to 1/2 MEMCLK for all lanes.
  3. Select two test addresses for each rank present. The addresses must be cache line (64 byte) aligned and separated by 2 MB starting with the first rank.
  4. Write one cache line where each byte is 55h to the first test address for each rank.
  5. Write one cache line where each byte is AAh to the second test address for each rank.
  6. For each lane, program the gross and fine timing fields in F2x[1, 0]9C\_x[2B:10] with a starting total delay value: For DDR2, the starting total delay value is zero. For DDR3, the starting value corresponds to the write DQS delays found during write leveling.
  7. For the start value and each subsequent total delay value in F2x[1, 0]9C\_x[2B:10] do the following:
    - A. For each rank:
      - a. Read the first test address for the current rank and compare each lane of the first data beat with each lane of the value written in step 5 above.
      - b. Reset the read pointer in the DRAM controller receive FIFO by writing the current corresponding DQS receiver enable delay settings to one of F2x[1, 0]9C\_x[2B:10].
      - c. Read the second test address for the rank and compare each lane of the first data beat with each lane of the value written in step 6 above.
      - d. Reset the read pointer in the DRAM controller receive FIFO by writing the current corresponding DQS receiver enable delay settings to one of F2x[1, 0]9C\_x[2B:10].
      - e. For each lane, save each DQS receiver enable setting that passes for both read patterns.
    - B. If the DIMM has two ranks then save each DQS receiver enable setting that passes for both ranks.
    - C. For each lane, update the current total DQS receiver enable delay setting in preparation for the next iteration using the following:
      - Save the total delay of a pass after a fail from a previous iteration as Delay1.
      - If the result of the current iteration is a fail, then add 1/8 UI to the total delay. If 1 MEMCLK of total delays have been tested without converging to a passing solution, then BIOS considers the channel untrainable and performs a vendor defined error recovery.
      - If the result of the current iteration is a pass then for each subsequent training iteration BIOS alternately adds either 1/32 UI or 1/4 UI to the total delay until either a fail condition prevails or until the lane is trained. If the current delay minus Delay1  $\geq$  1.25 UI (while no fail condition was detected) then the lane is trained; program F2x[1, 0]9C\_x[2B:10] = Delay1 + 7/8 UI and discontinue updating the delay value for the lane.
    - D. Swap the test address pointers for each rank, such that the order of data read on the bus alternates between 55h and AAh with each subsequent iteration of step 7. Continue until all lanes are trained.
  8. Flush the receiver FIFO with the following:
    - A. Write one cache line to any one of the test addresses on the DIMM, where each byte of the data in the

cache line is neither 55h nor AAh.

B. Read one cache line from the test address used above.

- Program F2x[1, 0]78[MaxRdLatency] with the current greatest value of F2x[1, 0]9C\_x[2B:10]. See 2.8.9.9.5 [Calculating MaxRdLatency].
- Program F2x[1, 0]78[DqsRcvEnTrain]=0.

### 2.8.9.9.3 DQS Position Training

DQS position training is used to place the DQS strobe in the center of the DQ data eye. Determining the correct DRAM DQS delay settings for both reads and writes must be performed using a two dimensional search of the read and write delay settings. This section describes the algorithm used to determine the values required to program the DRAM Write Data Timing registers (see F2x[1, 0]9C\_x[3:0]0[2:1] and F2x[1, 0]9C\_x[3:0]03) and the DRAM Read DQS Timing Control registers (see F2x[1, 0]9C\_x[3:0]0[6:5] and F2x[1, 0]9C\_x[3:0]07) registers for DQS position training.

To generate the needed continuous read or write data streams for DQS position training, see 2.8.9.9.6 [Continuous Pattern Generation].

1. Select three test addresses for each rank present in the system. The addresses must be cache line (64 byte) aligned. Fill all three addresses with cachelines of identical data for each byte location.
2. For each channel:
  - For each byte lane:
    - For each rank:
      - DRAM Write Data Timing Loop:**
        - For each DRAM Write Data Timing setting of the current byte:
          - Write the current write DQS delay value to the DRAM Write Data Timing register for the current byte lane.
          - Write the DRAM training pattern to the first test address for the rank.
        - DRAM Read DQS Timing Control Loop:**
          - For each read delay setting for the DRAM Read DQS Timing Control register:
            - a. Write the current DRAM Read DQS Timing Control delay setting for the current byte lane.
            - b. Read the DRAM training pattern from the first test address three times.
            - c. If the training pattern is read correctly, record the read position for the current byte lane as a pass; otherwise record the result as a fail.
            - d. Increment the DQS Read Timing Control Register setting for the current lane and continue in this **DRAM Read DQS Timing Control Loop**.
        - Process the array of results from step c above and determine the longest string of consecutive Read DQS values with passing results.
        - If the read DQ to DQS delay setting for the current byte lane contains three or more consecutive delay values with passing results, then exit the **DRAM Write Data Timing Loop** after programming the Read DQS Timing Control register with the average value of the smallest and largest values in the string of consecutive passing results.
        - Increment the Write DQS Timing Control Register byte for the current byte lane and continue the **DRAM Write Data Timing Loop**.
      - Write the Read DQS Timing Control register setting for the current byte with a value that represents the center position of the passing region.
      - Write 0 to the DRAM Write Data Timing register for the current byte lane.
      - For each DRAM Write Data Timing register setting:
        - a. Write the current DRAM DQS timing control register delay setting for the current byte

- lane.
  - b. Write 0's to the three test addresses for the current rank.
  - c. Write the DRAM training pattern to the three test addresses for the current rank.
  - d. Read the DRAM training pattern from the three test addresses.
  - e. If the training pattern is read correctly from each test address mark the DRAM Write Data Timing setting for the current byte lane as a pass.
  - f. Increment the DRAM DQS write timing register byte for the current byte lane and go to step a.
- Compare the passing regions for the current byte lane for each rank to determine a mutually centered region that passes for all ranks.
  - Write the Read DQS Timing Control register for the current byte lane with the centered delay setting of the mutually passing region for reads.
  - Write the DRAM Write Data Timing register for the current byte with the centered delay position of the mutual passing region for writes.

#### 2.8.9.4 ECC Byte Lane Training

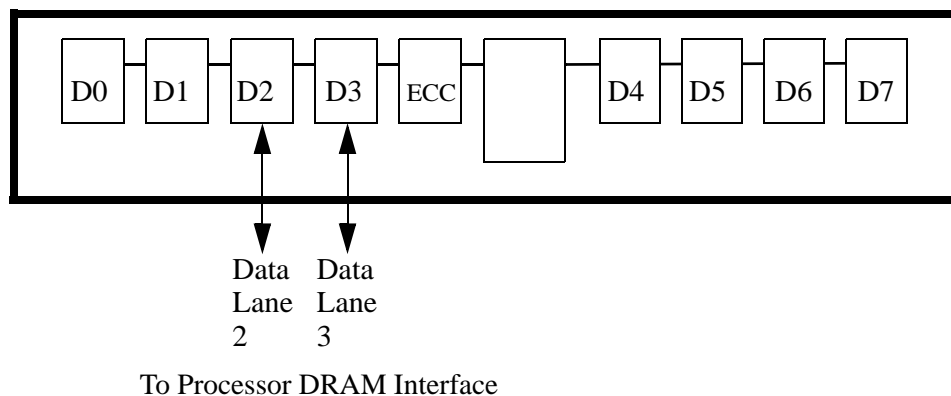
Because the ECC lanes of the DRAM interface are not visible to software additional steps are necessary in order to program the proper delay settings for the ECC lanes. Also, the different DDR3 DRAM layout topologies makes calculating the delay values of the ECC lanes problematical. However, in most cases, a simple averaging can be performed that yields reasonable delay values for the ECC lanes.

For DDR2 DIMMs, all ECC byte lane delay values can be determined by averaging the physically adjacent data lanes on the DRAM interface. For example, if the ECC byte lane signals on the board are physically between data byte lanes 3 and 4, the programmed value for the delay registers would be the average of the values used for data byte lanes 3 and 4. This technique would be used to calculate the delay settings for all DDR2 BIOS based training as mentioned in the previous paragraph.

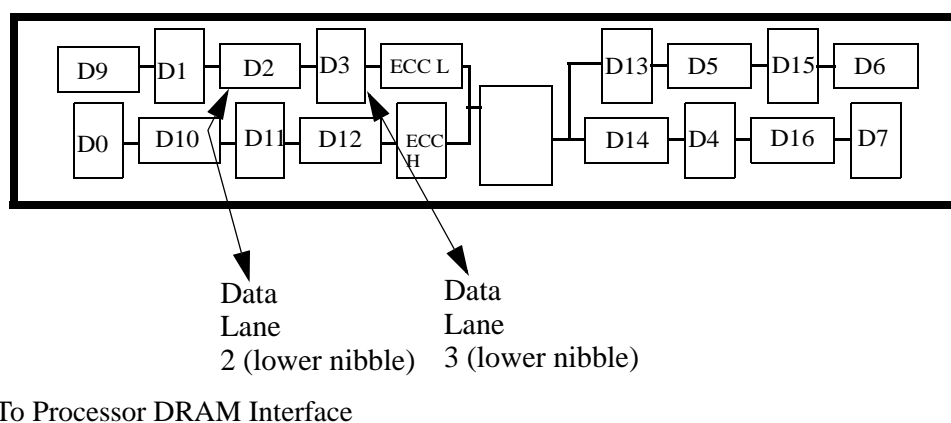
This method is also used to determine the ECC delay settings for unbuffered DDR3 DIMMs.

For DDR3 registered DIMMs, the ECC lane delay is calculated by extrapolating the trained delays of two adjacent lanes on the DIMM. See [Figure 11](#) for an example illustration. All data lanes on x4 DIMMs are trained using the lower nibbles by default so the data lanes chosen for the calculations are byte aligned. See [Figure 12](#) for an example illustration. The delay settings can be calculated using the following method:

- For each chip select of a 1 rank DIMM, or chip select pair of a 2 rank or 4 rank DIMM, BIOS calculates the needed delay value by subtracting the delay value used for data byte lane 2 from the value used for data byte lane 3. The difference could be either a negative or positive value.
- The difference is added to the delay setting used for data lane 3 and the result is used for the delay setting of the ECC lane.
- BIOS repeats this process for the other ranks as needed.



**Figure 11: DDR3 x8 Registered DIMM**



**Figure 12: DDR3 Registered x4 DIMM (planar)**

### 2.8.9.9.5 Calculating MaxRdLatency

The MaxRdLatency value determines when the node's 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 `F2x[1, 0]78[MaxRdLatency]`. The calculation is always done at NB P-state 0.

There are two processor specific delay values necessary for MaxRdLatency referred to as constants P and N in this section. P represents the portion of the delay in the DRAM clock domain. N represents the portion of the delay in the Northbridge clock domain. The values for P and N are defined as follows:

- For revision B, P = 5.5, N = 5.
- For revision C:
  - For F(1207), AM2r2, AM3, and S1g3 package types, use P = 5.5, N = 5.
  - Else use P=14.5, N= 2.
  - For revision D and later, P=14.5, N= 2.

The following steps describe the algorithm used to compute `F2x[1, 0]78[MaxRdLatency]` used for DRAM training. K is used as a temporary placeholder for the incrementally summed value.

1. Multiply the CAS Latency (in MEMCLKs) by 2 to get the number of 1/2 MEMCLKs units for Tc1 and store into K.
  - $K = 2 * CL$ ; See `F2x[1, 0]88[Tc1]`.
2. If registered DIMMs are used then add 2 to the incremental sub-total K.
  - If `F2x[1, 0]90[UnbuffDimm]=0` then  $K = K + 2$
3. If the all coarse prelaunch setup delays are 1/2 MEMCLK then add 1, else add 2 to the sub-total K.
  - If (`F2x[1, 0]9C_x04[AddrCmdSetup]` and `F2x[1, 0]9C_x04[CsOdtSetup]` and `F2x[1, 0]9C_x04[CkeSetup] = 0`) then  $K = K + 1$
  - If (`F2x[1, 0]9C_x04[AddrCmdSetup]` or `F2x[1, 0]9C_x04[CsOdtSetup]` or `F2x[1, 0]9C_x04[CkeSetup] = 1`) then  $K = K + 2$
4. If the `F2x[1, 0]78[RdPtrInit]` field is 4, 5, or 6, then add 4, 3, or 2, respectively, to the sub-total K.
  - $K = K + (8 - F2x[1, 0]78[RdPtrInit])$
5. Add the maximum (worst case) delay value of `F2x[1, 0]9C_x[2B:10][DqsRcvEnGrossDelay]` that exists across all DIMMs and byte lanes.
  - $K = K + (\text{Maximum } F2x[1, 0]9C_x[2B:10][DqsRcvEnGrossDelay])$
6. Add P to the sub-total K.
  - $K = K + P$
7. Convert the sub-total value K (in 1/2 MEMCLKs) to Northbridge clocks (NCLKs) normalized to 200 MHz clk (multiplying before dividing avoids rounding errors):
  - $K = K * 200 * (F3xD4[NbFid] + 4)$ ; see `F3xD4[NbFid]` for more information on the state of NbFid.
  - $K = K / (\text{current memory clock frequency})$ ; see `F2x[1, 0]88[MemClkFreq]`
  - $K = K / 2$ ; removes the 1/2 MEMCLK component
8. Add N to the sub-total.
  - $K = K + N$
9. Program the final MaxRdLatency with the total delay value (in NCLKs):
  - `F2x[1, 0]78[MaxRdLatency] = RoundUp(K)`

Note: if `F2x110[DctGangEn] = 1`, BIOS sets both DCT's `F2x[1, 0]78[MaxRdLatency]` to the maximum of either channel's computed MaxRdLatency value.

#### 2.8.9.9.5.1 MaxRdLatency Training

The following describes an algorithm that can be used to optimize `F2x[1, 0]78[MaxRdLatency]` value used after DRAM training:

The following three cache line pattern is used to train the MaxRdLatency value:

```
0C3C_FF52_6E0E_3FAC h
49C5_B613_4A68_8181 h
5C16_50E3_7C78_0BA6 h
0C67_53E6_0C4F_9D76 h
BABF_B6CA_2055_35A5 h
0C5F_1C87_610E_6E5F h
14C9_C383_4884_93CE h
9CE8_F615_F5B9_A5CD h
```

```
C38F_1B4C_AAD7_14B5 h
669F_7562_72ED_647C h
4A89_8B30_5233_F802 h
3326_B465_10A4_0617 h
C807_E3D3_5538_6E04 h
14B4_E63A_AB49_E193 h
```

EA51\_7C45\_67DF\_2495h  
F814\_0C51\_7624\_CE51h

B61D\_D0C9\_4824\_BD23h  
E8F3\_807D\_072B\_CFBEBh  
25E3\_0C47\_919E\_A373h  
4DA8\_0A5A\_FEB1\_2958h  
792B\_0076\_E9A0\_DDF8h  
F025\_B496\_E81C\_73DCh  
8085\_94FE\_1DB7\_E627h  
655C\_7783\_8266\_8268h

- For each channel:
  - If two channels are ganged then BIOS applies the worst case timing found in [\[The DRAM DQS Receiver Enable Timing Control Registers\] F2x\[1, 0\]9C\\_x\[2B:10\]](#) across both channels and performs the following actions once on both channels simultaneously.
  - BIOS calculates a starting MaxRdLatency delay value by executing steps 1 through 5, and 7 in section 2.8.9.9.5 above.
  - BIOS selects an address associated with the DIMM that has the worst case [\[The DRAM DQS Receiver Enable Timing Control Registers\] F2x\[1, 0\]9C\\_x\[2B:10\]](#) register setting that was found on the channel during DQS receiver enable training.
  - If two channels are ganged or if two unganged channels are interleaved with address bit 6 selecting the channel (see [F2x110\[DctSelIntLvAddr\]](#)), then write two copies of the specified pattern to the target address; otherwise write one copy of the specified pattern to the target address.
    1. Incrementing through all possible MaxRdLatency delay values beginning at the calculated MaxRdLatency start value:
    2. Set current MaxRdLatency delay value.
    3. Read three cache lines from the selected addresses on the current DIMM.
    4. Compare all three cache lines of data to the values written.
      - If the compare matches, go to step 5. below.
      - If the compare does not match, increment the MaxRdLatency value and go to step 2. above.
    5. Save the MaxRdLatency value for the current DIMM for the current channel.
    6. Repeat all the above steps for the other channel.
- Program the largest MaxRdLatency value in NCLKs plus 1 additional NCLK, plus 1 MEMCLK (to convert the MEMCLK value to NCLKs see 2.8.9.9.5 step 7 above) for each channel.

#### 2.8.9.9.6 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.

For reads, prefetch DRAM training mode is enabled through [\[The Memory Controller Configuration High Register\] F2x11C\[PrefDramTrainMode\]](#). In prefetch DRAM training mode, the DRAM prefetcher (see the same register) continues to issue to prefetches (once it detects a stride) until the DRAM prefetch limit, [F2x11C\[MctPrefReqLimit\]](#), is reached. This results in a series of back-to-back reads to the DCT; the corresponding data is stored in the prefetch data buffer. This data can then be accessed by subsequent reads to the strided addresses (and then checked for correctness by software). The expected sequence of events is as follows:

1. BIOS ensures that the only accesses outstanding to the MCT are training reads.



2. If `F2x[1, 0]90[BurstLength32]=1`, then BIOS ensures that the DCTs and DRAMs are configured for 64 byte bursts (8-beat burst length). See 2.8.4 [Burst Length]. This requires that BIOS issue MRS commands to the devices to change to an 8-beat burst length and then to restore the desired burst length after training is complete.
3. BIOS programs `F2x[1, 0]90[ForceAutoPchg] = 0` and `F2x[1, 0]8C[DisAutoRefresh] = 1`.
4. If necessary, BIOS programs `F2x[1, 0]78[EarlyArbEn] = 1` at this time. See register description.
5. BIOS sets `F2x11C[MctPrefReqLimit]` to the number of training reads (Ntrain) it wishes to generate in the training sequence.
6. BIOS sets `F2x11C[PrefDramTrainMode]` bit.
7. The act of setting `F2x11C[PrefDramTrainMode]` causes the MCT to flush out the prefetch stride predictor table (removing any existing prefetch stride patterns).
8. BIOS issues an SFENCE (or other serializing instruction) to ensure that the prior write completes.
9. For revision C and earlier processors, BIOS generates two training reads. For revision D and later processors BIOS generates three training reads. These must be to consecutive cache lines (i.e. 64 bytes apart) and must not cross a naturally aligned 4 KByte boundary.
10. These reads set up a stride pattern which is detected by the prefetcher. The prefetcher then continues to issue prefetches until `F2x11C[MctPrefReqLimit]` is reached, at which point the MCT clears `F2x11C[PrefDramTrainMode]`.
11. BIOS issues the remaining (Ntrain - 2 for revisions C and earlier or Ntrain - 3 for revision D and later) reads after checking that `F2x11C[PrefDramTrainMode]` is cleared. These reads must be to consecutive cache lines (i.e., 64 bytes apart) and must not cross a naturally aligned 4KB boundary. These reads hit the prefetches and read the data from the prefetch buffer.
12. When BIOS is ready to issue the next set of training reads, go to step #6.
13. When training is complete, BIOS disables the DRAM prefetcher training mode by programming `F2x11C[PrefDramTrainMode]=0`;
14. BIOS restores the target values for `F2x[1, 0]90[ForceAutoPchg]`, `F2x[1, 0]8C[DisAutoRefresh]` and `F2x[1, 0]90[BurstLength32]`.

For writes, prefetch DRAM training is accomplished using the write bursting function, described in `F2x11C`, as follows:

1. Disable the leaking of writes to the DCT that are below the burst watermark by setting `F2x11C[DctWrLimit] = 00b`.
2. Set `F2x11C[MctWrLimit]` to desired number of cachelines in the burst.
3. Flush out prior writes by setting `F2x11C[FlushWr]`.
4. Wait for `F2x11C[FlushWr]` to clear, indicating prior writes have been flushed.
5. Issue the stream of writes. When `F2x11C[MctWrLimit]` is reached (or when `F2x11C[FlushWr]` is set again), all the writes are written to DRAM.

### 2.8.9.10 DRAM Phy Power Savings

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

- Disable unused MEMCLK pins. See `F2x[1, 0]88[MemClkDis]`.
- Disable unused CKE, ODT, and chip select pins. See `F2x[1, 0]9C_x0C[CKETri, ODTTri, ChipSelTri]`.
- If a channel does not have memory then program `F2x[1, 0]88[DisDramInterface] = 1`.
- If the DIMMs do not have address parity, then program `F2x[1, 0]9C_x0D0F812F[TristateCA] = 1` for the PAR pin.

For revision DA-C and C3 and later:

- If the DCT does not have ECC memory, program `F2x[1, 0]9C_x0D0F0[F:0]30[PwrDn] = 1` for the ECC byte.
- If the DCT does not have x4 DRAMs, program `F2x[1, 0]9C_x0D0F0[F:0]13[RxDqsUDIIPowerDown] = 1`

for all byte lanes.

- For each chiplet instance (a chiplet instance controls the pads for two clocks) where all pads are unused by the package, the platform, or the DIMM, program `F2x[1,0]9C_x0D0F2[F:0]30[PwrDn] = 1`.

## 2.8.10 Memory Interleaving Modes

Interleaving is defined as the spreading contiguous physical address space over multiple DIMM banks, as opposed to each DIMM owning a single contiguous address space. This is accomplished by using lower-order address bits to select between DIMMs. The processor supports three different types of interleaving modes:

- CS: interleaving between the DIMM banks of a channel based the CS. This is controlled through [\[The DRAM CS Base Address Registers\] F2x\[1, 0\]\[5C:40\]](#).
- Channel: interleaving between the two 64-bit channels of a processor. This is controlled through [\[The DRAM Controller Select Low Register\] F2x110\[DctSelIntLvEn\]](#).
- Node: interleaving between DIMMs of different processor nodes. This is controlled through [\[The DRAM Base/Limit Registers\] F1x\[1, 0\]\[7C:40\]](#) and [\[The DRAM Limit System Address Register\] F1x124](#). See 2.8.10.2 [\[Node Interleaving\]](#).

Any combination of these interleaving modes may be enabled concurrently. It is recommended that interleaving between chip selects is enabled if the configuration supports it.

### 2.8.10.1 Chip Select Interleaving

The chip select memory interleaving mode requires all DIMM chip-select ranges be the same size and type, and the number of chip selects a power of two. A BIOS algorithm for programming [\[The DRAM CS Base Address Registers\] F2x\[1, 0\]\[5C:40\]](#) and [\[The DRAM CS Mask Registers\] F2x\[1, 0\]\[6C:60\]](#) in memory interleaving mode is as follows:

1. Program all DRAM CS Base Address and DRAM CS Mask registers using contiguous normalized address mapping.
2. For each enabled chip select, swap the corresponding BaseAddr[38:27] bits with the BaseAddr[21:13] bits as defined in [Table 68](#) and [Table 69](#) for DDR2 and [Table 70](#) and [Table 71](#) for DDR3.
3. For each enabled chip select, swap the corresponding AddrMask[38:27] bits with the AddrMask[21:13] bits as defined in [Table 68](#) and [Table 69](#) for DDR2 and [Table 70](#) and [Table 71](#) for DDR3.

**Table 68. DDR2 swapped normalized address lines for interleaving for a 64-bit interface**

Chip Select Mode	Chip Select Size	Swapped Base Address and Address Mask bits		
		8 way CS interleaving	4 way CS interleaving	2 way CS interleaving
0000b	128-MB	[29:27] and [16:14]	[28:27] and [15:14]	[27] and [14]
0001b	256-MB	[30:28] and [17:15]	[29:28] and [16:15]	[28] and [15]
0010b	512-MB	[31:29] and [17:15]	[30:29] and [16:15]	[29] and [15]
0011b	512-MB	[31:29] and [18:16]	[30:29] and [17:16]	[29] and [16]
0100b	512-MB	[31:29] and [18:16]	[30:29] and [17:16]	[29] and [16]
0101b	1-GB	[32:30] and [18:16]	[31:30] and [17:16]	[30] and [16]
0110b	1-GB	[32:30] and [18:16]	[31:30] and [17:16]	[30] and [16]
0111b	2-GB	[33:31] and [18:16]	[32:31] and [17:16]	[31] and [16]
1000b	2-GB	[33:31] and [19:17]	[32:31] and [18:17]	[31] and [17]

**Table 68. DDR2 swapped normalized address lines for interleaving for a 64-bit interface**

Chip Select Mode	Chip Select Size	Swapped Base Address and Address Mask bits		
		8 way CS interleaving	4 way CS interleaving	2 way CS interleaving
1001b	4-GB	[34:32] and [19:17]	[33:32] and [18:17]	[32] and [17]
1010b	4-GB	[34:32] and [18:16]	[33:32] and [17:16]	[32] and [16]
1011b	8-GB	[35:33] and [19:17]	[34:33] and [18:17]	[33] and [17]

**Table 69. DDR2 swapped normalized address lines for CS interleaving for a 128-bit interface**

Chip Select Mode	Chip Select Size	Swapped Base Address and Address Mask bits		
		8 way CS interleaving	4 way CS interleaving	2 way CS interleaving
0000b	256-MB	[30:28] and [17:15]	[29:28] and [16:15]	[28] and [15]
0001b	512-MB	[31:29] and [18:16]	[30:29] and [17:16]	[29] and [16]
0010b	1-GB	[32:30] and [18:16]	[31:30] and [17:16]	[30] and [16]
0011b	1-GB	[32:30] and [19:17]	[31:30] and [18:17]	[30] and [17]
0100b	1-GB	[32:30] and [19:17]	[31:30] and [18:17]	[30] and [17]
0101b	2-GB	[33:31] and [19:17]	[32:31] and [18:17]	[31] and [17]
0110b	2-GB	[33:31] and [19:17]	[32:31] and [18:17]	[31] and [17]
0111b	4-GB	[34:32] and [19:17]	[33:32] and [18:17]	[32] and [17]
1000b	4-GB	[34:32] and [20:18]	[33:32] and [19:18]	[32] and [18]
1001b	8-GB	[35:33] and [20:18]	[34:33] and [19:18]	[33] and [18]
1010b	8-GB	[35:33] and [19:17]	[34:33] and [18:17]	[33] and [17]
1011b	16-GB	[36:34] and [20:18]	[35:34] and [19:18]	[34] and [18]

**Table 70. DDR3 swapped normalized address lines for interleaving for a 64-bit interface**

Chip Select Mode	Chip Select Size	Swapped Base Address and Address Mask bits		
		8 way CS interleaving	4 way CS interleaving	2 way CS interleaving
0001b	256-MB	[30:28] and [18:16]	[29:28] and [17:16]	[28] and [16]
0010b	512-MB	[31:29] and [18:16]	[30:29] and [17:16]	[29] and [16]
0101b	1-GB	[32:30] and [18:16]	[31:30] and [17:16]	[30] and [16]
0110b	1-GB	[32:30] and [19:17]	[31:30] and [18:17]	[30] and [17]
0111b	2-GB	[33:31] and [18:16]	[32:31] and [17:16]	[31] and [16]
1000b	2-GB	[33:31] and [19:17]	[32:31] and [18:17]	[31] and [17]
1001b	4-GB	[34:32] and [19:17]	[33:32] and [18:17]	[32] and [17]
1010b	4-GB	[34:32] and [18:16]	[33:32] and [17:16]	[32] and [16]
1011b	8-GB	[35:33] and [19:17]	[34:33] and [18:17]	[33] and [17]

**Table 71. DDR3 swapped normalized address lines for CS interleaving for a 128-bit interface**

Chip Select Mode	Chip Select Size	Swapped Base Address and Address Mask bits		
		8 way CS interleaving	4 way CS interleaving	2 way CS interleaving
0001b	512-MB	[31:29] and [19:17]	[30:29] and [18:17]	[29] and [17]
0010b	1-GB	[32:30] and [19:17]	[31:30] and [18:17]	[30] and [17]
0101b	2-GB	[33:31] and [19:17]	[32:31] and [18:17]	[31] and [17]
0110b	2-GB	[33:31] and [20:18]	[32:31] and [19:18]	[31] and [18]
0111b	4-GB	[34:32] and [19:17]	[33:32] and [18:17]	[32] and [17]
1000b	4-GB	[34:32] and [20:18]	[33:32] and [19:18]	[32] and [18]
1001b	8-GB	[35:33] and [20:18]	[34:33] and [19:18]	[33] and [18]
1010b	8-GB	[35:33] and [19:17]	[34:33] and [18:17]	[33] and [17]
1011b	16-GB	[36:34] and [20:18]	[35:34] and [19:18]	[34] and [18]

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

- The register settings for contiguous memory mapping are:
  - F2x[1, 0]80 = 0000\_0011h // CS0/1 = 256 MB; CS2/3 = 256 MB
  - F2x[1, 0]40 = 0000\_0001h // 0 MB base
  - F2x[1, 0]44 = 0010\_0001h // 256 MB base = 0 MB + 256 MB
  - F2x[1, 0]48 = 0020\_0001h // 512 MB base = 256 MB + 256 MB
  - F2x[1, 0]4C = 0030\_0001h // 768 MB base = 512 MB + 256 MB
  - F2x[1, 0]60 = 0008\_3FE0h // CS0/CS1 = 256 MB
  - F2x[1, 0]64 = 0008\_3FE0h // CS2/CS3 = 256 MB
- The base address bits to be swapped are defined in Table 70, 256MB chip select size, 4 way CS interleaving column. The BaseAddr[29:28] bits are specified by F2x[1, 0][5C:40][BaseAddr[21:20]]. The BaseAddr[17:16] bits are specified by F2x[1, 0][5C:40][BaseAddr[9:8]].
  - F2x[1, 0]40 = 0000\_0001h
  - F2x[1, 0]44 = 0000\_0101h
  - F2x[1, 0]48 = 0000\_0201h
  - F2x[1, 0]4C = 0000\_0301h
- The AddrMask bits to be swapped are the same as the BaseAddr bits defined in the previous step. The AddrMask[29:28] bits are specified by F2x[1, 0][6C:60][AddrMask[21:20]]. The AddrMask[17:16] bits are specified by F2x[1, 0][6C:60][AddrMask[9:8]].
  - F2x[1, 0]60 = 0038\_3CE0h
  - F2x[1, 0]64 = 0038\_3CE0h

### 2.8.10.2 Node Interleaving

If node interleaving is enabled, then (1) all nodes in the system must contain the same amount of DRAM, (2) all the DRAM of all nodes in the system must be interleaved, and (3) the base and limit registers for all nodes must be programmed to 0 and top of memory, respectively. If node interleaving and channel interleaving are enabled, all DRAM channels in the system must have the same amount of DRAM.

Node interleaving for up to 8 nodes is controlled by F1x[1, 0][7C:40][IntlvEn and IntlvSel],

**F1x120**[DramIntlvSel] and **F1x124**[DramIntlvEn]. IntlvEn and DramIntlvEn are programmed to specify the interleaved address bits (programmed the same in each node). IntlvSel specifies the value that those address bits need to be to target a node (must be programmed to a different value for each node). DramIntlvSel specifies the value of those address bits for the local node. It is expected that one [The DRAM Base/Limit Registers] **F1x[1, 0][7C:40]** pair is enabled per node; one of these pairs selects the local node by having an IntlvSel value that matches **F1x120**[DramIntlvSel]; IntlvEn is the same in all [The DRAM Base/Limit Registers] **F1x[1, 0][7C:40]** pairs and the same as **F1x124**[DramIntlvEn] of all nodes. For example, a 4-node system would normally be programmed as follows for interleaving:

<b>Node 0</b> - IntlvEn = 00_0011b - IntlvSel = 000b - Addr[13:12] = 00b	<b>Node 1</b> - IntlvEn = 00_0011b - IntlvSel = 001b - Addr[13:12] = 01b
<b>Node 2</b> - IntlvEn = 00_0011b - IntlvSel = 010b - Addr[13:12] = 10b	<b>Node 3</b> - IntlvEn = 00_0011b - IntlvSel = 011b - Addr[13:12] = 11b

### 2.8.11 Memory Hoisting

Memory hoisting is defined as reclaiming the DRAM space that would naturally reside in the MMIO hole just below the 4G address level. This memory is repositioned above the 4G level when the registers that control memory hoisting, [The DRAM Hole Address Register] **F1xF0**, [The DRAM Controller Select Low Register] **F2x110**, [The DRAM Controller Select High Register] **F2x114**, are set up properly.

The memory hoisting offset fields, **F1xF0**[DramHoleOffset] and **F2x114**[DctSelBaseOffset], are programmed based on the following parameters:

- **F1xF0**[DramHoleBase], which is the base address of the IO hole below the 4G level. In MP systems, this should be programmed to the same value in all processors.
- **F2x110**[DctSelBaseAddr], which specifies the base address of the upper memory space owned by one of the DCTs.
- **F2x110**[DctSelIntLvEn], which specifies if interleaving between the two DCTs is enabled (channel interleave mode).
- **F1x120**[DramBaseAddr], and **F1x124**[DramLimitAddr], which specify the address range of the node.
- If both DCTs are enabled (**F2x[1, 0][5C:40]**[CSEnable]). Note: if the two DCTs are ganged in 128-bit mode, then only 1 DCT is defined to be enabled in the case conditions below.

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

### 2.8.11.1 DramHoleOffset Programming

F1xF0[DramHoleOffset] is programmed to one of the following equations based on the scenario:

- **Case 1:** if only one DCT is enabled OR both DCTs are enabled in channel interleaved mode and have equal amount of memory OR DctSelBaseAddr > DramHoleBase, then:  

$$\text{DramHoleOffset}[31:23] = \{\text{DramHoleSize}[31:24], 0b\} + \{\text{DramBaseAddr}[31:27], 0000b\};$$
- **Case 2:** if both DCTs are enabled in channel non-interleaved mode and DctSelBaseAddr < DramHoleBase, then:  

$$\text{DramHoleOffset}[31:23] = \{\text{DramHoleSize}[31:24], 0b\} + \{\text{DctSelBaseAddr}[31:27], 0000b\};$$
- **Case 3:** if both DRAM controllers are enabled in channel interleaved mode and DctSelBaseAddress < DramHoleBase, then:  

$$\text{DramHoleOffset}[31:23] = \{\text{DramHoleSize}[31:24], 0b\} + \{\text{DramBaseAddr}[31:27], 0000b\} \\ + \{0b, (\text{DctSelBaseAddr}[31:27] - \text{DramBaseAddr}[31:27]), 000b\};$$

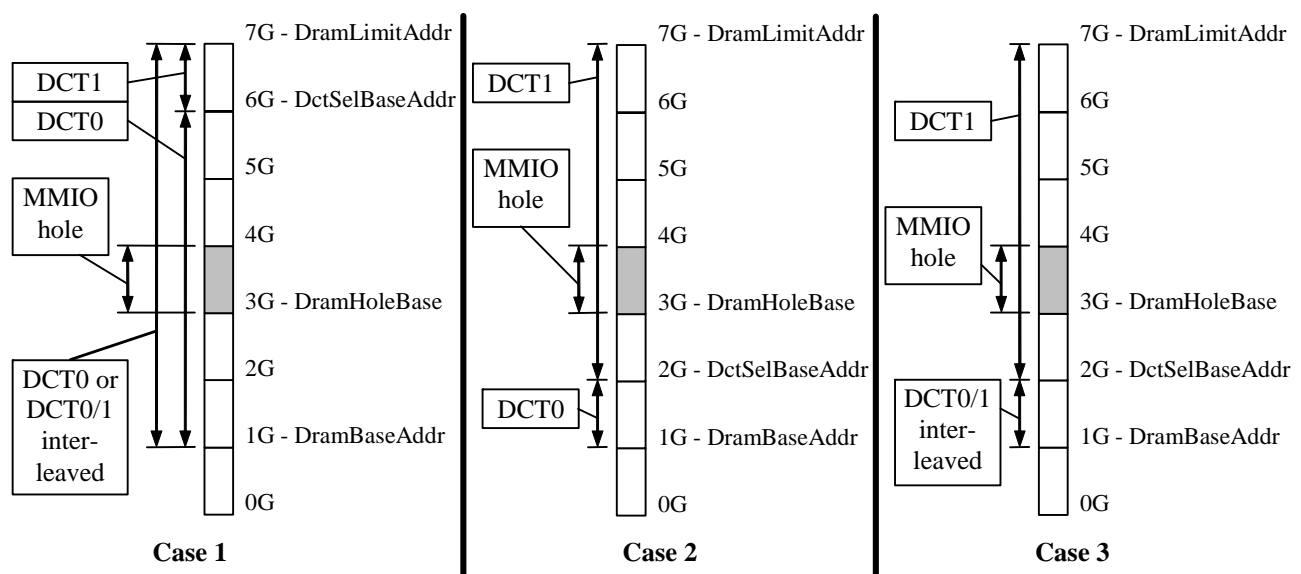


Figure 13: Example cases for programming DramHoleOffset.

### 2.8.11.2 DctSelBaseOffset Programming

F2x114[DctSelBaseOffset] is programmed to one of the following equations based on the scenario:

- **Case 1:** if the two DCTs are enabled in channel non-interleaved mode, then:  

$$\text{DctSelBaseOffset}[47:26] = \{\text{DctSelBaseAddr}[47:27], 0b\};$$
- **Case 2:** if (1) the two DCTs are enabled in channel interleaved mode and DctSelBaseAddr < DramHoleBase OR if (2) there is no memory hole in the address map, then:  

$$\text{DctSelBaseOffset}[47:26] = \{\text{DramBaseAddr}[47:27], 0b\} \\ + \{0b, (\text{DctSelBaseAddr}[47:27] - \text{DramBaseAddr}[47:27])\};$$
- **Case 3:** if the two DCTs are enabled in channel interleaved mode, DctSelBaseAddr > DramHoleBase, and the interleaved range includes the MMIO hole, then:  

$$\text{DctSelBaseOffset}[47:26] = \{\text{DramBaseAddr}[47:27], 0b\} \\ + \{0000h, \text{DramHoleSize}[31:26]\} \\ + \{0b, (\text{DctSelBaseAddr}[47:27] \\ - \{0000h, (\text{DramBaseAddr}[31:27] + \text{DramHoleSize}[31:27])\})\};$$

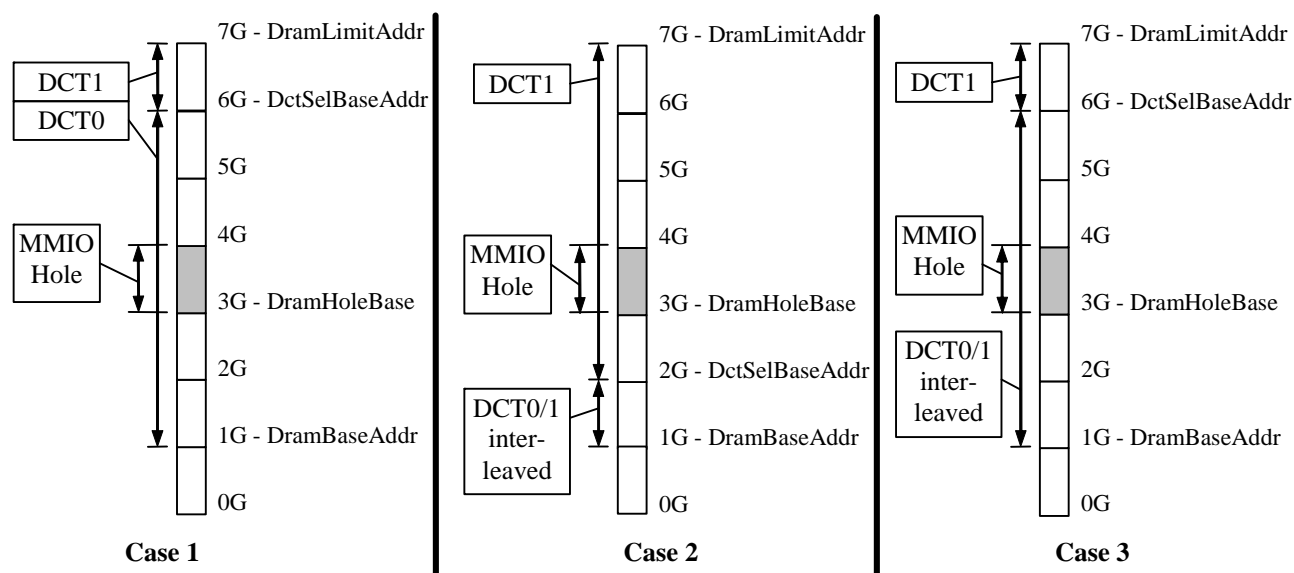


Figure 14: Example cases for programming DctSelBaseOffset.

## 2.8.12 On-Line Spare

On-line spare is a RAS mechanism that allows the system to reserve one rank of one logical DIMM to be used as a spare rank. System software reserves a spare rank by setting  $F2x[1, 0][5C:40][Spare]$  in one of the CS Base address registers. The spare rank must be greater than or equal to the size of all other ranks in the system.

The system can switch to the spare rank when system software determines that one of the ranks in the system is no longer functioning properly and needs to be replaced. The on-line spare mechanism is controlled by [\[The On-Line Spare Control Register\] F3xB0](#). System software initiates the swap to the spare rank by writing the chip select number of the bad rank to  $F3xB0[BadDramCS]$  and setting  $F3xB0[SwapEn]$ .

On-line spare is not supported in UMA systems.

### 2.8.12.1 On-Line Spare and CS Interleaving

The on-line spare feature can only be used with 2 way and 4 way CS interleaving under the following conditions.

- All ranks of each DIMM present must be of the same size and configuration.
- Only the following populations are supported:
  - 2 DIMMs per channel (2 way CS interleaving)
    - One single rank DIMM and one dual rank DIMM. Any rank can be used as the spare rank.
    - Both DIMMs are dual rank. Any rank can be used as the spare rank. One rank must be marked as bad since only two ranks can be active.
  - 3 DIMMs per channel (4 way CS interleaving)
    - Two dual rank DIMMs and one single rank DIMM. Any rank can be used as the spare rank.
    - All DIMMs are dual rank. Any rank can be used as the spare rank. One rank must be marked as bad since only four ranks can be active.
  - 4 DIMMs per channel (4 way CS interleaving)
    - One dual rank DIMMs and three single rank DIMMs. Any rank can be used as the spare rank.
    - Two dual rank DIMMs and two single rank DIMMs. Any rank can be used as the spare rank. One



rank must be marked as bad since only four ranks can be active.

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

For revision D and later processors, each DCT can throttle commands based on the state of the processor's EVENT\_L pin (one per controller) or when  $F2x[1, 0]A4[BwCapEn]=1$ . The EVENT\_L pin is used for thermal management and  $F2x[1, 0]A4[BwCapEn]$  is used to limit memory power independent of thermal management.

JEDEC defines two DRAM device types: Standard (with a case temperature of 85C), and extended temperature (with a case temperature of 95C). For extended temperature devices, a minimum refresh rate of 3.9 us is required when the temperature exceeds 85C. The recommended usage and the interaction between DRAM case temperature and EVENT\_L pin throttling is as follows:

- The BIOS may enable command throttling on a DRAM controller if the platform supports the EVENT\_L pin by programming  $F2x[1, 0]A4[ODTSEn] = 1$ .
  - The recommended usage is for this pin to be connected to one or more JEDEC defined on DIMM temperature sensor(s). The SPD of the DIMM indicates whether the DIMM supports on DIMM temperature sensor.
  - 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  $F2x[1, 0]A4[CmdThrottleMode]$  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. See  $F2x[1, 0]A4[ODTSEn]$ .
- If all DIMMs support extended temperature range (specified in the DIMMs' SPD ROM), the 3.9 us refresh rate requirement can be satisfied by one of the following methods:
  - BIOS programs the refresh rate to 3.9 us in  $F2x[1, 0]8C[Tref]$ .
  - BIOS configures the temperature sensor trip point for all DIMMs according to the 85 degrees C case temperature specification.
- Standard and extended temperature devices may be mixed in a system. In this case, BIOS has two options:
  - BIOS programs the refresh rate to 3.9 us in  $F2x[1, 0]8C[Tref]$ , or,
  - BIOS configures the temperature sensor trip point for all DIMMs according to the 85 degrees C case temperature specification of the standard temperature DIMM(s).
- 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 latched status of the EVENT\_L pin for both DCTs can be read by system software in  $F2xAC[MemTempHot1, MemTempHot0]$ .

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

- The trip points for each DIMM are 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.

## 2.9 CPU Core

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

### 2.9.1 Virtual Address Space

The processor supports 48 address bits of virtual memory space (256 terabyte) as indicated by **CPUID Fn8000\_0008\_EAX**.

### 2.9.2 CPU Cores and Downcoring

Each node supports 1, 2, 3, 4, 5, or 6 cores as follows:

- The number of cores supported by the node is specified by **F3xE8[CmpCap]**.
- Cores may be *downcored* (removed) by **F3x190[DisCore[5:0]]**. Based on **F3xE8[CmpCap]**, **DisCore[0]** applies to a single-core node; **DisCore[1:0]** apply to a dual-core node; **DisCore[2:0]** apply to a 3-core node; **DisCore[3:0]** apply to a 4-core node; **DisCore[4:0]** apply to a 5-core node; **DisCore[5:0]** apply to a 6-core node.
- **F3x190[DisCore]** affects **CPUID Fn8000\_0008\_ECX[NC]**.
- Software is required to use **F3x190[DisCore[5:0]]** as follows:
  - 1, 2, 3, 4, 5, or 6 cores must be enabled on each node (0-core configurations are not allowed).
  - For dual-node processors, the number of cores enabled in each internal node must be the same. See **F3xE8[IntNodeNum]** for how to identify the internal nodes in a dual-node processor. The specific cores enabled in each internal node do not need to be the same, only the total number of cores in each internal node.
  - BIOS should configure all processors in a system to have the same number of enabled cores.
  - Setting bits corresponding to cores that are not present results in undefined behavior.
  - Once a core has been removed, it cannot be added back without a cold reset.
  - If the number of cores in the system is changed, then **F0x60[CpuCnt[4:0]]** and **F0x160[CpuCnt[7:5]]** in all nodes must be updated to reflect the new value after the warm reset.
  - BIOS should configure **MSRC001\_102A[ThrottleNbInterface]** to reflect the number of enabled cores.
- The core number, *CpuCoreNum*, is provided to SW running on each core through **CPUID Fn0000\_0001\_EBX[LocalApicId]** and **APIC20[ApicId]**, formatted based on the state of **MSRC001\_001F[InitApicIdCpuIdLo]**; *CpuCoreNum* also affects **F0x68[Cpu1En]** and **F0x168[Cpu5En, Cpu4En, Cpu3En and Cpu2En]**. *CpuCoreNum*, varies as the lowest integers from 0 to 5, 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.
- In dual-node processors, the *CpuCoreNum* assignments are described below. NC is the number of cores in the processor as specified by **CPUID Fn8000\_0008\_ECX[NC]**.
  - For the BSP, cores 0 through (NC+1)/2-1 are assigned to the node where **F0x60[NodeId]=0h**. Cores (NC+1)/2 through NC are assigned to the node where **F0x60[NodeId]=1h**. NC is specified by **CPUID Fn8000\_0008\_ECX[NC]**.
  - For APs, cores 0 through (NC+1)/2-1 are assigned to internal node 0 and cores (NC+1)/2 through NC are assigned to internal node 1. **F3xE8[IntNodeNum]** specifies the internal node number.
- In dual-node processors, **MSRC001\_001F[InitApicIdCpuIdLo]** provides **F0x60[NodeId[2:1]]**. The full *NodeId* is provided in **MSRC001\_100C**.

Some legacy operating systems do not support three core or six core processors. The BIOS should support a user configurable option to disable one core in a three core processor or two cores in a six core processor for legacy operating system support.

### 2.9.3 Access Type Determination

The access type determination and destination affects routing specified in section 2.6.4 [Northbridge Routing].

### 2.9.3.1 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 an IO 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 the CPU must determine before issuing the access to the NB: the cache attribute (e.g., WB, WC, UC; as described in the MTRRs) and the access destination (DRAM or MMIO).

#### 2.9.3.1.1 Determining The Cache Attribute

1. The CPU translates the logical address to a physical address. In that process it determines the initial cache attribute based on the settings of the Page Table Entry PAT bits, [The MTRR Default Memory Type Register (MTRRdefType)] MSR0000\_02FF, [The Variable-Size MTRRs (MTRRphysBasen and MTRRphys-Maskn)] MSR0000\_02[0F:00], and [The Fixed-Size MTRRs (MTRRfixn)] MSR0000\_02[6F:68, 59, 58, 50].
2. The ASeg and TSeg SMM mechanisms are then checked in parallel to determine if the initial cache attribute should be overridden (see [The SMM TSeg Base Address Register (SMMAddr)] MSRC001\_0112 and [The SMM TSeg Mask Register (SMMMask)] MSRC001\_0113). If the address falls within an enabled ASeg/TSeg region, then the final cache attribute is determined as specified in MSRC001\_0113.

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

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

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

1. (Lowest priority) Compare against the top-of memory (TOM) registers (see MSRC001\_001A, and MSRC001\_001D).
2. The IORRs (see MSRC001\_00[18, 16] and MSRC001\_00[19, 17]).
3. [The Fixed-Size MTRRs (MTRRfixn)] MSR0000\_02[6F:68, 59, 58, 50].
4. TSEG & ASEG (see MSRC001\_0112 and MSRC001\_0113).
5. (Highest priority) NB AGP aperture range registers.

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

1. The CPU compares the address against [The Top Of Memory Register (TOP\_MEM)] MSRC001\_001A, and [The Top Of Memory 2 Register (TOM2)] MSRC001\_001D, to determine if the default access destination is DRAM or MMIO space.
2. The CPU then compares the address against the IORRs (MSRC001\_00[18, 16] and MSRC001\_00[19, 17]); if it matches, the default access destination is overridden as specified by the IORRs. BIOS can use the IORRs to create an IO hole within a range of addresses that would normally be mapped to DRAM. It can also use the IORRs to re-assert a DRAM destination for a range of addresses that fall within a bigger IO hole that overlays DRAM. Some key points to consider:
  - a) Operating system software never needs to program IORRs to re-map addresses that naturally target DRAM; any such programming is done by the BIOS.
  - b) The IORRs should not cover the range used for the AGP aperture if the GART logic in the NB is enabled.

- c) The IORRs should be programmed to cover the AGP aperture if the aperture/GART translation is handled by an IO device (e.g., the chipset).
3. For addresses below 1M byte, the address is then compared against the appropriate Fixed MTRRs to override the default access destination. Each fixed MTRR includes two bits, RdDram and WrDram, that determine the destination based on the access type. See [MSR0000\\_02\[6F:68, 59, 58, 50\]](#).
4. The ASeg and TSeg SMM mechanisms are then checked in parallel to determine if the destination should be overridden (see [MSRC001\\_0112](#) and [MSRC001\\_0113](#)). If the address falls within an enabled ASeg/TSeg region, then the destination is determined as specified in [MSRC001\\_0113](#).

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

Note: BIOS must ensure that when it makes IO cacheable, IO devices in the cacheable region will respond correctly to cacheable requests. If this requirement cannot be met, BIOS must protect these IO regions from cacheable requests. The recommended method is to make them not cacheable.

## 2.9.4 Timers

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

- [\[The Time Stamp Counter Register \(TSC\)\] MSR0000\\_0010](#); the TSC increments at the rate specified by [MSRC001\\_0015\[TscFreqSel\]](#).
- The APIC timer ([APIC380](#) and [APIC390](#)), which counts at the rate of CLKIN; the APIC timer may count in units of between 1 and 8. During C1E, the APIC timer is imprecise if [F3x80\[NbLowPwrEn\]=1](#) and [F3xD4\[NbClkDiv\]](#) is non-zero.

## 2.9.5 APIC

### 2.9.5.1 ApicId Enumeration Requirements

System hardware and BIOS must ensure that the number of cores per processor (NC) exposed to the operating system by all tables, registers, and instructions across all cores and processors in the system is identical. See [2.15.1 \[Multi-Core Support\]](#) to derive NC.

Operating systems are expected to use [CPUID Fn8000\\_0008\\_ECX\[ApicIdCoreIdSize\[3:0\]\]](#), the number of least significant bits in the Initial APIC ID that indicate core ID within a processor, in constructing per-core CPUID masks. ([ApicIdCoreIdSize\[3:0\]](#) determines the maximum number of cores (MNC) that the processor could theoretically support, not the actual number of cores that are actually implemented or enabled on the processor, as indicated by [CPUID Fn8000\\_0008\\_ECX\[NC\]](#).) BIOS must use the ApicId MNC rule when assigning [\[The APIC ID Register\] APIC20\[ApicId\]](#) values as described below.

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

$$\text{ApicId}[\text{proc}=i, \text{core}=j] = (\text{OFFSET\_IDX} + i) * \text{MNC} + j$$

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

It is recommended that BIOS use the following APIC ID assignments for the broadest operating system support. Given  $N = (\text{Number\_Of\_Processors} * \text{MNC})$  and  $M = \text{Number\_Of\_IOAPICs}$ :

- If  $(N+M) < 16$ , assign the local (core) ApicId's first from 0 to  $N-1$ , and the IOAPIC IDs from  $N$  to  $N+(M-1)$ .
- If  $(N+M) \geq 16$ , assign the IOAPIC IDs first from 0 to  $M-1$ , and the local (core) ApicId's from  $K$  to  $K+(N-1)$ , where  $K$  is an integer multiple of  $MNC$  greater than  $M-1$ .

For example, consider a 3 processor system where each processor has 3 cores and there are 8 IOAPIC devices. Each core can support an 8-bit ApicId. But if each IOAPIC device supports only a 4-bit IOAPIC ID, then the problem can be solved by shifting the CPU ApicId space to start at some integer multiple of  $MNC$ , such as offset 8 ( $MNC = 4$ ;  $OFFSET\_IDX=2$ ):

$ApicId[proc=0,core=0] = (2+0)*4 + 0 = 0x08$	$ApicId[proc=1,core=2] = (2+1)*4 + 2 = 0x0E$
$ApicId[proc=0,core=1] = (2+0)*4 + 1 = 0x09$	$ApicId[proc=2,core=0] = (2+2)*4 + 0 = 0x10$
$ApicId[proc=0,core=2] = (2+0)*4 + 2 = 0x0A$	$ApicId[proc=2,core=1] = (2+2)*4 + 1 = 0x11$
$ApicId[proc=1,core=0] = (2+1)*4 + 0 = 0x0C$	$ApicId[proc=2,core=2] = (2+2)*4 + 2 = 0x12$
$ApicId[proc=1,core=1] = (2+1)*4 + 1 = 0x0D$	

## 2.10 Thermal Functions

Thermal functions HTC, STC and THERMTRIP are intended to maintain processors temperature in a valid range by:

- Providing an input to the external circuitry that controls cooling.
- Lowering power consumption by switching to lower-performance P-state or.
- 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. The processor includes a thermal diode as well.

### 2.10.1 The Tctl Temperature Scale

Tctl is the processor temperature control value, used by the platform to control cooling systems. Tctl is accessible through SB-TSI and [F3xA4\[CurTmp\]](#). Tctl is a non-physical temperature on an arbitrary scale measured in degrees. It does *not* represent an actual physical temperature like die or case temperature. Instead, it specifies the processor temperature relative to the point at which the system must supply the maximum cooling for the processor's specified maximum case temperature and maximum thermal power dissipation. It is defined as follows for all parts:

- For  $Tctl = 0$  to  $Tctl\_max - 0.125$ : the temperature of the part is  $[Tctl\_max - Tctl]$  degrees under the temperature for which maximum cooling is required.
- For  $Tctl = Tctl\_max$  to  $255.875$ : the temperature of the part is  $[Tctl - Tctl\_max]$  degrees over the worst-case expected temperature under normal conditions. The processor may take corrective actions that affects performance or operation as a result, such as invoking HTC or THERMTRIP\_L.

### 2.10.2 Thermal Diode

The thermal diode is a diode connected to the THERMDA and THERMDC pins used for thermal measurements. External devices use measurements from the thermal diode measurements to calculate temperature during operation and test. These measurements are required to be adjusted as specified by [F3xE4\[DiodeOffset\]](#). This diode offset supports temperature sensors using two sourcing currents only. Other sourcing current implementations are not compatible with the diode offset and are not supported. A correction to the offset may be required for temperature sensors using other current sourcing methods. Contact the temperature sensor vendor

to determine whether an offset correction is needed. Feature support varies by package. See the *Infrastructure Roadmap*.

### 2.10.3 Temperature-Driven Logic

The temperature calculated by the TCC is used by HTC, STC, THERMTRIP, and the PROCHOT signal.

#### 2.10.3.1 PROCHOT\_L and Hardware Thermal Control (HTC)

The processor *HTC-active state* is characterized by (1) the assertion of PROCHOT\_L, (2) reduced power consumption, and (3) reduced performance. While in the HTC-active state, the processor reduces power consumption by limiting all cores to a P-state (specified by F3x64[HtcPstateLimit]). See 2.4.2 [P-states]. While in the HTC-active state, software should not change F3x64 (except for HtcActSts and HtcEn). 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 F3x64.

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 15 ns.

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

- F3xE8[HtcCapable]=1
- F3x64[HtcEn]=1
- PWROK=1
- THERMTRIP\_L=1
- The processor is not in the C3 ACPI state.

and any of the following conditions are true:

- Tctl is greater than or equal to the HTC temperature limit (F3x64[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 (F3x64[HtcTmpLmt]).
- Tctl has become less than the HTC temperature limit (F3x64[HtcTmpLmt]) minus the HTC hysteresis limit (F3x64[HtcHystLmt]) since being greater than or equal to the HTC temperature limit (F3x64[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 Software Thermal Control (STC)

STC is controlled by [The Software Thermal Control (STC) Register] F3x68. This register provides a software-controlled mechanism to alter power consumption based on temperature. When the processor control temperature (Tctl; see 2.10.1 [The Tctl Temperature Scale]) exceeds the temperature threshold specified by F3x68[StcTmpLmt], then the processor enters the *STC thermal zone*. When it subsequently drops below F3x68[StcTmpLmt] minus F3x68[StcHystLmt], the processor exits the STC thermal zone. F3x68 controls whether interrupts or special bus cycles (which may be converted into interrupts by the chipset) are generated when the processor transitions into and out of the STC thermal zone. The interrupt handler may take an action to alter power consumption or alter the level of external cooling.



One way that software may reduce power is to program the processor to enter the *STC-active state*. This is like the *HTC-active state*, however `PROCHOT_L` is not asserted. The processor enters the *STC-active state* if `F3x68[StcPstateEn]=1`. While in the *STC-active state*, the processor limits the performance to the P-state specified by [\[The Software Thermal Control \(STC\) Register\] F3x68\[StcPstateLimit\]](#); See 2.4.2 [\[P-states\]](#).

### 2.10.3.3 THERMTRIP

If the processor supports the *THERMTRIP* state (as specified by [\[The Thermtrip Status Register\] F3xE4\[ThermtpEn\]](#) or `CPUID Fn8000_0007[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.
- In addition, the external chipset is expected to place the system into the *S5 ACPI state* (power off) if `THERMTRIP_L` is detected to be asserted.

A cold reset is required to exit the *THERMTRIP* state.

## 2.11 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 24 to 31, where device 24 corresponds to node 0 and device 31 corresponds to node 7. See 2.11.3 [\[Processor Configuration Space\]](#) for more information on 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 [\[The IO-Space Configuration Address Register\] IOCF8\[ConfigEn\]](#), which allows access to BCS.
  - Access to ECS enabled through [\[The Northbridge Configuration Register \(NB\\_CFG\)\] MSRC001\\_001F\[EnableCf8ExtCfg\]](#).
  - Only PCI-defined segment 0 is accessible.
  - Use of IO-space configuration can be programmed to generate GP faults through [\[The Hardware Configuration Register \(HWCR\)\] MSRC001\\_0015\[IoCfgGpFault\]](#).
  - SMI trapping for these accesses is specified by [\[The IO Trap Control Register \(SMI\\_ON\\_IO\\_TRAP\\_CTL\\_STS\)\] MSRC001\\_0054](#) and [\[The IO Trap Registers \(SMI\\_ON\\_IO\\_TRAP\\_\[3:0\]\)\] MSRC001\\_00\[53:50\]](#).
- MMIO configuration: configuration space is a region of memory space.
  - The base address and size of this range is specified by [\[The MMIO Configuration Base Address Register\] MSRC001\\_0058](#). The size is controlled by the number of configuration-space bus numbers supported by the system. Accesses to this range are converted to configuration space accesses as follows:
    - `Address[31:0] = {0h, bus[7:0], device[4:0], function[2:0], 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.

See [2.6.4.1.3 \[Configuration Space\]](#) for details on configuration space routing.

### 2.11.1 MMIO Configuration Coding Requirements

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

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

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

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

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

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

### 2.11.2 MMIO Configuration Ordering

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

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

### 2.11.3 Processor Configuration Space

The processor includes configuration space as described in section [3 \[Registers\]](#). Accesses to unimplemented registers of implemented functions are ignored: writes dropped; reads return 0's. Accesses to unimplemented functions are 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 non-existent processors (e.g., device 25 of a single-node system) are routed based on the configuration map registers. If such requests are master aborted, then the processor can log the event.

## 2.12 RAS and Advanced Server Features

This section applies reliability, availability, and serviceability, or RAS, and related advanced server considerations.

### 2.12.1 Machine Check Architecture

The processor contains logic and registers to detect, log, and (if possible) correct errors in the data or control paths in each core and the Northbridge.

Refer to the *AMD64 Architecture Programmer's Manual* for an architectural overview and methods for determining the processor's level of MCA support. See 1.2 [Reference Documents].

### 2.12.1.1 Machine Check Registers

The presence of the machine check registers is indicated by `CPUID Fn[8000_0001, 0000_0001]_EDX[MCA]`. The ability of hardware to generate a machine check exception upon an error is indicated by `CPUID Fn[8000_0001, 0000_0001]_EDX[MCE]`.

The machine check register set includes:

- Global status and control registers:
  - [The Global Machine Check Capabilities Register (MCG\_CAP)] `MSR0000_0179`
  - [The Global Machine Check Status Register (MCG\_STAT)] `MSR0000_017A`
  - [The Global Machine Check Exception Reporting Control Register (MCG\_CTL)] `MSR0000_017B`
- Most of the machine check MSRs are organized as a 4-register-type by 6-register-bank matrix.
  - The four register types are:
    - **MCi\_CTL**, The Machine Check Control Register: Enables error reporting via machine check exception (MCE). The `MCi_CTL` register in each bank must be enabled by the corresponding enable bit in `MCG_CTL (MSR0000_017B)`.
    - **MCi\_STATUS**: The Machine Check Status Register: Logs information associated with errors.
    - **MCi\_ADDR**: The Machine Check Address Register: Logs address information associated with errors.
    - **MCi\_MISC**: The Machine Check Miscellaneous Registers: Log miscellaneous information associated with errors, as defined by each error type.
  - The six error-reporting register banks supported are:
    - **MC0, DC**: `MSR0000_04[03:00]`, data cache machine check registers.
    - **MC1, IC**: `MSR0000_04[07:04]`, instruction cache machine check registers.
    - **MC2, BU**: `MSR0000_04[0B:08]`, bus unit machine check registers.
    - **MC3, LS**: `MSR0000_04[0F:0C]`, load-store machine check registers.
    - **MC4, NB**: `MSR0000_04[13:10]`, Northbridge machine check registers. The NB MC registers also include `MSRC000_04[0A:08]`. These MSRs are accessible from configuration space as well.
    - **MC5, FR**: `MSR00000_04[17:14]`, fixed-issue reorder buffer machine check registers.

Once system software has determined that machine check registers exist via the `CPUID` instruction, `MSR0000_0179` may be read to determine how many machine check banks are implemented and if [The Global Machine Check Exception Reporting Control Register (MCG\_CTL)] `MSR0000_017B` is present.

Table 72 identifies the addresses associated with each MCA register.

**Table 72: MCA register cross-reference table**

Register Bank (MCi)	MCA Register				
	CTL	STATUS	ADDR	MISC	CTL_MASK <code>MSRC001_00[49:44]</code>
MC0	<code>MSR0000_0400</code>	<code>MSR0000_0401</code>	<code>MSR0000_0402</code>	<code>MSR0000_0403</code>	<code>MSRC001_0044</code>
MC1	<code>MSR0000_0404</code>	<code>MSR0000_0405</code>	<code>MSR0000_0406</code>	<code>MSR0000_0407</code>	<code>MSRC001_0045</code>
MC2	<code>MSR0000_0408</code>	<code>MSR0000_0409</code>	<code>MSR0000_040A</code>	<code>MSR0000_040B</code>	<code>MSRC001_0046</code>
MC3	<code>MSR0000_040C</code>	<code>MSR0000_040D</code>	<code>MSR0000_040E</code>	<code>MSR0000_040F</code>	<code>MSRC001_0047</code>
MC4	<code>MSR0000_0410</code>	<code>MSR0000_0411</code>	<code>MSR0000_0412</code>	<code>MSR0000_0413</code> <code>MSRC000_04[0A:08]</code>	<code>MSRC001_0048</code>

**Table 72: MCA register cross-reference table**

Register Bank (MCi)	MCA Register				
	CTL	STATUS	ADDR	MISC	CTL_MASK MSRC001_00[49:44]
MC5	MSR0000_0414	MSR0000_0415	MSR0000_0416	MSR0000_0417	MSRC001_0049

Correctable and uncorrectable errors that are enabled in MCi\_CTL are logged in MCi\_STATUS and MCi\_ADDR as they occur. Uncorrectable errors immediately result in a Machine Check exception. For the NB, some errors only increment a counter in MC4\_MISC, which may trigger an interrupt (see 2.12.1.6 [Error Thresholding]).

Each MCi\_CTL register must be enabled by the corresponding enable bit in [The Global Machine Check Exception Reporting Control Register (MCG\_CTL)] MSR0000\_017B.

Additionally, [The Machine Check Control Mask Registers (MCi\_CTL\_MASK)] MSRC001\_00[49:44] allow BIOS to mask the presence of any error source enables from software for test and debug. When error sources are masked, it is as if the error was not detected. Such masking consequently prevents error responses.

Each register bank implements a number of machine check miscellaneous registers, denoted as MCi\_MISCj, where j goes from 0 to a maximum of 8. The presence of valid information in the first MCi\_MISC register (MCi\_MISC0) is indicated by MCi\_STATUS[MiscV], and in subsequent registers by MCi\_MISCj[Valid]. If there is more than one MCi\_MISC register in a given bank, a non-zero value in MCi\_MISC0[BlkPtr] points to the contiguous block of additional registers.

### 2.12.1.2 Machine Check Errors

There are two classes of machine check errors defined:

- Correctable: errors that can be corrected by hardware or microcode and cause no loss of data or corruption of processor state.
- Uncorrectable: errors that cannot be corrected by hardware or microcode and may have caused the loss of data or corruption of processor state.

Correctable errors are always corrected (unless disabled by implementation-specific bits in control registers for test or debug reasons). If they are enabled for logging, the status and address registers in the corresponding register bank are written with information that identifies the source of the error.

Uncorrectable errors, if enabled for logging, update the status and address registers, and if enabled for reporting, cause a machine check exception. If there is information in the status and address registers from a previous correctable error, it is overwritten. If an uncorrectable error is not enabled for logging, the error is ignored.

The implications of the two main categories of errors are (shown with a non-exhaustive list of examples):

1. Corrected error; the problem was dealt with.
  - Operationally (error handling), no action needs to be taken, because program flow is unaffected.
  - Diagnostically (fault management), software may collect information to determine if any components should be de-configured or serviced.
  - Examples include:
    - Correctable ECC, corrected online.
2. Uncorrected error; the problem was not dealt with.
  - Operationally (error handling), action does need to be taken, because program flow is affected.

- Diagnostically (fault management), software may collect information to determine if and what components should be de-configured or serviced.
- Examples include:
  - Uncorrectable ECC, no way to avoid passing it to process.

Machine check conditions can be simulated by using [MSRC001\\_0015](#)[McStatusWrEn]. This is useful for debugging machine check handlers.

For revision D and later, [F3x180](#)[ChgUcToCeEn] can be used to convert NB uncorrectable errors to correctable errors for debug observability.

### 2.12.1.2.1 Machine Check Error Logging and Reporting

An error is considered enabled for logging if all of the following are true:

- The global enable for the corresponding error-reporting bank in [[The Global Machine Check Exception Reporting Control Register \(MCG\\_CTL\)](#)] [MSR0000\\_017B](#) is set to 1.
- The corresponding mask bit for the error in [[The Machine Check Control Mask Registers \(MCi\\_CTL\\_MASK\)](#)] [MSRC001\\_00](#)[49:44] is cleared to 0.

An error is considered enabled for reporting if all of the following are true:

- The error is enabled for logging.
- 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 enabled for reporting.
- CR4.MCE is enabled (and no other model-specific configuration settings prevent machine check exception); see *AMD64 Architecture Programmer's Manual Volume 2: System Programming*.

Notes:

1. If error reporting is enabled but CR4.MCE is disabled, a reportable error will cause the system to enter shutdown.
2. If error reporting is disabled, the setting of CR4.MCE has no effect.

### 2.12.1.2.2 Error Overflow

During error overflow conditions (see [MSR0000\\_0401](#)[Over] and [MSR0000\\_0411](#)[Over]), an error which has already been logged in the status register may be overwritten.

[Table 73](#) indicates which errors are overwritten in the MC0 and MC4 error status registers. [Table 74](#) indicates which errors are overwritten in the MC1, MC2, MC3, and MC5 error status registers.

**Table 73: MC0 and MC4 Overwrite Priorities**

			Older Error			
			Uncorrectable		Correctable	
			Enabled	Disabled	Enabled	Disabled
Younger Error	Uncorrectable	Enabled	-	Overwrite	Overwrite	Overwrite
		Disabled	-	Overwrite	Overwrite	Overwrite
	Correctable	Enabled	-	Overwrite	Overwrite	Overwrite
		Disabled	-	Overwrite	Overwrite	Overwrite

**Table 74: MC1, MC2, MC3, and MC5 Overwrite Priorities**

			Older Error			
			Uncorrectable		Correctable	
			Enabled	Disabled	Enabled	Disabled
Younger Error	Uncorrectable	Enabled	-	Overwrite	Overwrite	Overwrite
		Disabled	-	Overwrite	Overwrite	Overwrite
	Correctable	Enabled	-	Overwrite	-	Overwrite
		Disabled	-	Overwrite	-	Overwrite

### 2.12.1.3 MCA Initialization

For the fundamentals of MCA initialization, refer to *AMD64 Architecture Programmer's Manual Volume 2: System Programming*. (See 1.2 [Reference Documents].)

Enabling error reporting: It is normally the responsibility of the operating system to enable error reporting via CR4.MCE, MCG\_CTL, and the MCi\_CTL registers (see 2.12.1.1 [Machine Check Registers]).

Masking error logging: When necessary, BIOS should mask undesired error logging via the MCi\_CTL\_MASK registers (MSRC001\_00[49:44]).

See 2.3.1 [BSP initialization] and 2.3.2 [AP initialization] for additional requirements based on reset type.

A BIOS that wants to ensure reliable operation in the event that a machine check occurs during POST may perform the requirements outlined in *AMD64 Architecture Programmer's Manual Volume 2: System Programming*, writing MCG\_CTL with all ones to enable logging, and writing zeros into each MCi\_CTL to prevent reporting. 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 POST 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. Alternatively, the BIOS may perform the requirements outlined in *AMD64 Architecture Programmer's Manual Volume 2: System Programming* except for 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).

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.

### 2.12.1.4 Handling Machine Check Exceptions

Machine check exceptions are used to report hardware errors to software.

At a minimum, the machine check handler must be capable of logging errors for later examination. The handler should log as much information as is needed to diagnose the error.

More thorough exception handler implementations can analyze errors to determine if each error is recoverable at the system level. 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 by hardware, but may still be recoverable

by software. Also 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.

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 the 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 each core. The status registers are numbered from 0 to one less than the value found in [MSR0000\\_0179](#)[Count]. For example, if the Count field indicates five status registers are supported, they are numbered MC0\_STATUS to MC4\_STATUS.
    - Check the valid bit in each status register (MCi\_STATUS[Val]). The remainder of the MCi\_STATUS register does not need to be examined when its valid bit is clear.
    - When identifying the error condition, portable exception handlers should examine MCi\_STATUS[Error Code] and [ErrorCodeExt].
    - 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 MCi\_STATUS registers to see if error recovery is possible. Error recovery is not possible when:
    - The processor context corrupt indicator (MCi\_STATUS[PCC]) is set to 1.
    - The error overflow status indicator (MCi\_STATUS[Over]) is set to 1. This indicates that more than one machine check error has occurred, but only one error is reported by the status register. This condition precludes recovery only when status for an uncorrectable error (UC) has been lost. See [2.12.1.2.2 \[Error Overflow\]](#) for situations in which UC error status can be lost by being overwritten. If the configuration is such that UC errors cannot be overwritten, then the overflow condition is not fatal.
  - If error recovery is not possible, the handler should log the error information and return to the operating system.
  - Check MCi\_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 [MSR0000\\_017A](#)[RIPV] is set, the interrupted program can be restarted reliably at the instruction pointer address pushed onto the exception handler stack. If RIPV is clear, the interrupted program cannot be restarted reliably, although it may be possible to restart it for debugging purposes.
  - When logging errors 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 guaranteed to be related to the error.
- 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, 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.

A machine check exception cannot be relied on to stop the propagation of bad data. A sync flood should be used for this purpose where recommended.

### 2.12.1.5 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 de-allocation, de-configuration, 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 also the unit in error. The next simplest form of indictment is *eyewitness indictment*, where the part in error 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 part in error 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 the physical location in error, such as a DRAM address or a cache index. Logical addresses, while identifying the data, do not identify the location of the data.

If possible, physical storage locations in DRAM or 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 uncorrectable errors are hard and correctable errors are soft. Repeated correctable errors from the same location are an indication that the fault is actually hard.

#### 2.12.1.5.1 Common Diagnosis Information

A common set of diagnosis information is useful for many problems. [Table 75](#) 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 mis-configuration.

**Table 75: MCA Registers Commonly Used for Diagnosis**

MCA Bank	Status	Configuration
MC0	MSR0000_0401 MSR0000_0402 MSR0000_0403	MSR0000_0400 MSRC001_0044
MC1	MSR0000_0405 MSR0000_0406 MSR0000_0407	MSR0000_0404 MSRC001_0045



**Table 75: MCA Registers Commonly Used for Diagnosis**

MCA Bank	Status	Configuration
MC2	MSR0000_0409 MSR0000_040A MSR0000_040B	MSR0000_0408 MSRC001_0046
MC3	MSR0000_040D MSR0000_040E MSR0000_040F	MSR0000_040C MSRC001_0047
MC4	MSR0000_0411 MSR0000_0412 MSR0000_0413 MSRC000_04[0A:08] F2xAC	MSR0000_0410 MSRC001_001F MSRC001_0048 F3x44 F3xE4 F3xE8 F3x180
MC5	MSR0000_0415 MSR0000_0416 MSR0000_0417	MSR0000_0414 MSRC001_0049

If examining MCA registers after startup, determine the cause of the startup:

- INIT; F0x6C[InitDet].
- Cold reset; F0x6C[ColdRstDet].
- Warm reset; if not INIT or cold reset.

To see if a link failure occurred, examine F0x[E4, C4, A4, 84][LinkFail]. If set, look for additional information:

- A sync flood saves a status of Sync Error in MC4\_STATUS.
- CRC error saves a status of CRC Error in MC4\_STATUS. See F0x[E4, C4, A4, 84][CrcErr and CrcFloodEn].
- Link not present does not save status in MC4\_STATUS. See F0x[E4, C4, A4, 84][InitComplete].

Other registers may be needed depending on the specific error symptoms.

### 2.12.1.6 Error Thresholding

For some types of errors, the hardware maintains counts of correctable and uncorrectable errors. When the counter reaches a programmable threshold, an event may optionally be triggered to inform software. This is known as error thresholding. The primary purpose of error thresholding is to help software recognize an excessive rate of correctable errors, which may be used as an indicator of marginal or failing hardware. This information can be used to make decisions about de-configuring hardware or scheduling service actions.

The error thresholding hardware reports 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. Furthermore, 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 the following error threshold groups as identified in Table 93 on page 296. Note

that for all error threshold groups, some number of correctable 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.

- DRAM
  - Memory errors can be counted and reported via [MSR0000\\_0413](#).
  - Operating systems can avoid using memory pages with excessive errors.
  - Spare memory can dynamically replace memory with excessive errors. See [2.8.12 \[On-Line Spare\]](#).
- Links
  - Link errors can be counted and reported via [MSRC000\\_0408](#) (see [MSRC000\\_04\[0A:08\]](#)).
  - 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 [2.7 \[Links\]](#) for details and restrictions on configuring links.
- L3 cache
  - L3 cache errors can be counted and reported via [MSRC000\\_0409](#) (see [MSRC000\\_04\[0A:08\]](#)).

### 2.12.1.7 Scrub Rate Considerations

This section gives guidelines for the scrub rate settings available in [\[The Scrub Rate Control Register\] F3x58](#). Scrubbers are used to periodically read cacheline sized data locations and associated tags, correcting any correctable errors which are discovered before they can migrate into uncorrectable 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. Scrubbers also help identify marginal or failed hardware by finding and logging repeated errors at the same locations (see also [2.12.1.6 \[Error Thresholding\]](#)).

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

For steady state operation, finding a range of useful scrub rates may be done 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 below baselines are made to maximize error coverage without affecting performance and not based on specific processor soft error rates. Refer to JEDEC standards for guidelines on adjusting for geographic location.

**Table 76: Recommended Scrub Rates per Node**

Register	Memory Size per Node (GB)	Register Setting	Scrub Rate
<a href="#">F3x58[DcacheScrub]</a>	-	12h	5.24 ms
<a href="#">F3x58[L2Scrub]</a>	-	10h	1.31 ms
<a href="#">F3x58[L3Scrub]</a>	-	10h	1.31 ms

**Table 76: Recommended Scrub Rates per Node**

Register	Memory Size per Node (GB)	Register Setting	Scrub Rate
F3x58[DramScrub]	0	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 <= 8GB	0Fh	655.4 us
	8 GB < size <= 16 GB	0Eh	327.7 us
	16 GB < size	0Dh	163.8 us

For low power states in which the processor core is halted, the power management configuration may affect scrubbing; see 2.6.6 [Memory Scrubbers] for details

### 2.12.1.8 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:

- Links: [F0x14C:130](#), [F3x44](#)

Error simulation involves creating the appearance to software that an error occurred. This is done by manually setting the MCA registers with desired values (see [MSRC001\\_0015](#)[McStatusWrEn]), and then driving the software via INT18. McStatusWrEn can be used to debug machine check interrupt handlers. 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 (INT $n$  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.

### 2.12.2 DRAM Considerations for ECC

DRAM is protected against errors by an error correcting code (ECC). The capabilities of the ECC depend on how the ECC symbols are mapped to the physical DRAM devices.

There are two different error correcting codes supported by the memory controller. Both DRAM error correcting codes are 128/16 (data bits/check bits) BCH codes. The primary difference between them is the symbol size; one code uses 8-bit symbols, and the other uses 4-bit symbols. A symbol is a group of bits which are symbol-size aligned; the x8 code uses 8-bit symbols which are 8-bit aligned, and the x4 code uses 4-bit symbols which are 4-bit aligned. For the x8 code, bits 0–7 make symbol 0, bits 8–15 make symbol 1, and so on. For the x4 code, bits 0–3 make symbol 0, bits 4–7 make symbol 1, and so on. For Revision C and earlier revisions, only the x4 code is available. For Revision D and later, both x4 and x8 codes are available, and one must be selected during configuration by setting [F3x180](#)[EccSymbolSize].

A single symbol error is any bit error combination within one symbol. Both of these ECC codes are able to detect and correct any number of incorrect bits in a single symbol, to detect any number of incorrect bits in two separate symbols, and may detect errors in more than two symbols depending on the position of corrupted symbols. (The x4 code is able to detect 100% of errors in two symbols. The x8 code is able to detect 99.99999963% of errors in two symbols.)

ECC has different characteristics depending on the physical configuration of the memory, including DRAM device width, ganged vs. unganged DRAM modes, and multi-beat interleaving.

- DRAM device width refers to the number of bits sourced simultaneously from a single memory chip. For example, a x4 DRAM device provides 4 bits in a single beat.
- Ganged refers to the use of both DRAM controllers within a memory controller acting in concert to access memory. Unganged mode uses only a single DRAM controller for each memory access, and therefore reads multiple beats from the same DRAM devices. See [2.8.5 \[Ganged or Unganged Mode Considerations\]](#).
- Multi-beat interleaving refers to the way bits from the different memory beats are organized to form an ECC line. See [2.12.2.1 \[Multi-Beat Interleaving for Unganged Mode\]](#).

In certain configurations (see [Table 77](#)), the ECC provides “chipkill” functionality; all single symbol errors caused by a failed DRAM device are corrected. When a DRAM device fails, the code is able to correct the entire lost symbol, as long as there are no other symbols with errors. In cases where the symbol size is smaller than the DRAM device width, DRAM device failures result in multiple symbol errors, and cannot be corrected.

For an indication of the current hardware settings regarding chipkill, see [F3x44\[ChipKillEccCap\]](#).

**Table 77: Configuration Specific Recommended ECC Symbol Size**

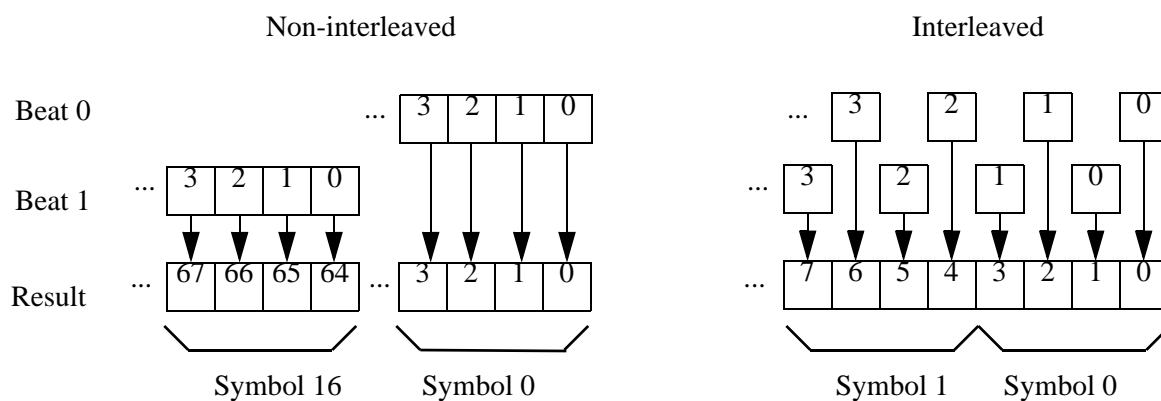
Revision	DRAM Width <sup>1</sup>	DCT Mode <sup>2</sup>	Recommended Symbol Size <sup>3</sup>	Error Description
C and earlier	x4	Unganged	x4	The failure of a DRAM device results in errors to two symbols, and can be detected but cannot be corrected.
		Ganged		The failure of a DRAM device results in an error to a single symbol and can be corrected (chipkill).
	Unganged	> x4		The failure of a DRAM device results in errors to multiple symbols, and can be detected with a high probability but cannot be corrected.
		Ganged		The failure of a DRAM device results in errors to multiple symbols and cannot be corrected. Failure of a x8 device can be detected with 100% probability, since exactly two symbols are affected.

**Table 77: Configuration Specific Recommended ECC Symbol Size**

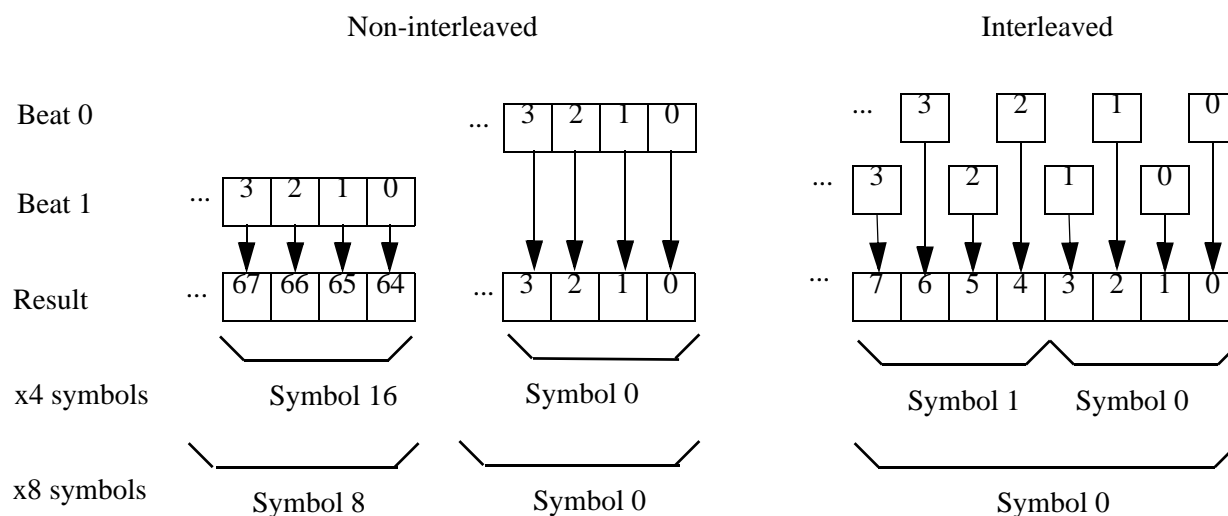
Revision	DRAM Width <sup>1</sup>	DCT Mode <sup>2</sup>	Recommended Symbol Size <sup>3</sup>	Error Description
D	x4	Unganged	x8	The failure of a DRAM device results in errors to one symbol, and can be corrected (chipkill).
		Ganged	x4	The failure of a DRAM device results in an error to a single symbol and can be corrected (chipkill).
	x8	Unganged	x8	The failure of a DRAM device results in errors to two symbols, and can be detected but cannot be corrected.
		Ganged		The failure of a DRAM device results in errors to a single symbol, and can be corrected (chipkill).
	> x8	Unganged/ Ganged		The failure of a DRAM device results in errors to multiple symbols, and can be detected with a high probability but cannot be corrected.
	1. See F2x[1, 0]90[X4Dimm]. 2. See F2x110[DctGangEn]. 3. See F3x180[EccSymbolSize]. DRAM devices wider than the symbol size are not recommended for high reliability or high availability systems, due to the higher potential for uncorrectable, undetected, or miscorrected errors.			

**2.12.2.1 Multi-Beat Interleaving for Unganged Mode**

Unganged DRAM mode uses only a single DCT for each memory access, and therefore requires two beats of 64-bit wide DRAM data from the same DRAM devices to form the 128-bit wide data upon which to apply the ECC function. When specified by F2x110[DctDatIntLv], even and odd bits from the two 64-bit data beats can be interleaved to create the 128-bit line as shown in Figure 15 and Figure 16.



**Figure 15: Example of line interleaving from x4 DRAM in unganged DRAM mode using x4 ECC**



**Figure 16: Example of line interleaving from x4 DRAM in ungangged DRAM mode using x8 ECC**

When multi-beat interleaving is performed in ungangged DRAM mode, a partially failing device (e.g., pin failure) contributes two incorrect bits to the same symbol, which can be corrected by the ECC. A totally failing DRAM device (i.e., chip failure) which is wide enough to contribute error bits to two different symbols results in an uncorrectable error.

When using the ECC syndrome to find the bits in error, see [Figure 15](#) and [Figure 16](#) to map from the symbol and bit number back to the correct device bit. Refer to section [2.8.6 \[Routing DRAM Requests\]](#) for details on how to map to a DIMM and device.

### 2.12.2.2 Software Managed Bad Symbol Identification

x8 ECC is augmented by a software managed algorithm to provide additional coverage against two symbol errors. Bad symbol identification is managed by software and used by hardware to improve detection of a second symbol in error in the same beat.

When software determines that a DRAM device is bad, it should program the rank and symbol into the BadDramCs and BadDramSymbol fields of the appropriate [\[The DCT0 Bad Symbol Identification Register\] F3x138](#) or [\[The DCT1 Bad Symbol Identification Register\] F3x13C](#). The bad symbol is determined by using the syndrome captured during correctable errors, as described in section [2.12.2.3.1 \[x8 ECC\]](#). Determining that a device is bad can be done using the guidelines in section [2.12.1.6 \[Error Thresholding\]](#).

The memory controller uses the bad symbol identification registers during fetches to determine if errors are correctable or uncorrectable. When the bad symbol identification register contains valid information:

- If a fetched line has an error only in the same rank and symbol identified by this register, then the error is a correctable chipkill error.
- If a fetched line has an error in the same rank and a different symbol than identified by this register, then there are at least two symbol errors in the line and the error is uncorrectable.

### 2.12.2.3 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 F3x180[EccSymbolSize].

For correctable errors, the DIMM in error is uniquely identified by the error address (F3x50[ErrAddr]) and the ECC syndrome (F3x48[Syndrome[15:8]] and F3x4C[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.

### 2.12.2.3.1 x8 ECC

The use of x8 ECC is indicated in F3x180[EccSymbolSize].

The syndrome field uniquely identifies the failing bit positions of a correctable ECC error. Only syndromes identified in Table 78 are correctable by the error correcting code.

Symbols 00h-07h map to data bits 0-63; symbols 08h-0Fh map to data bits 64-127; symbol 10h maps to special purpose non-data bits; symbols 11h-12h map to ECC check bits for data bits 0-127 and symbol 10h.

To use Table 78, find the 16-bit syndrome value in the table. Because of the large size of the table, this is most easily done by taking the low order byte of the syndrome as the row number, then scanning the row for the complete 16-bit syndrome. If it is not found, use the high order byte of the syndrome as the row number, then scan the row for the complete 16-bit syndrome. Once the syndrome is found in the table, the corresponding Symbol In Error column indicates which symbol, and therefore which DIMM has the error. The Error Bitmask column indicates the bits in error in the symbol. To map to the DIMM, use the algorithm in section 2.8.6 [Routing DRAM Requests].

For example, assume the ECC syndrome is 03EAh. First search row EAh for the complete syndrome. Since it is not found, search row 03h for the complete syndrome. It is found in column 9h, so symbol 9h has the error. Since the error bitmask indicates value 3h (0011b), bits 0 and 1 within that symbol are corrupted. Symbol 9h maps to bits 72-79, so the corrupted bits are 72 and 73 of the line.

**Table 78: x8 ECC correctable syndromes**

Error Bit-mask	Symbol In Error																		
	12h	11h	10h	Fh	Eh	Dh	Ch	Bh	Ah	9h	8h	7h	6h	5h	4h	3h	2h	1h	0h
00	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
01	0100	0001	0101	01B8	015C	012E	01C6	0163	01FD	0189	019D	B801	5C01	2E01	C601	6301	FD01	8901	9D01
02	0200	0002	0202	0201	02B8	025C	02FD	02C6	028B	0263	024B	0102	B802	5C02	FD02	C602	8B02	6302	4B02
03	0300	0003	0303	03B9	03E4	0372	033B	03A5	0376	03EA	03D6	B903	E403	7203	3B03	A503	7603	EA03	D603
04	0400	0004	0404	0402	0401	04B8	048B	04FD	0467	04C6	0496	0204	0104	B804	8B04	FD04	6704	C604	9604
05	0500	0005	0505	05BA	055D	0596	054D	059E	059A	054F	050B	BA05	5D05	9605	4D05	9E05	9A05	4F05	0B05
06	0600	0006	0606	0603	06B9	06E4	0676	063B	06EC	06A5	06DD	0306	B906	E406	7606	3B06	EC06	A506	DD06
07	0700	0007	0707	07BB	07E5	07CA	07B0	0758	0711	072C	0740	BB07	E507	CA07	B007	5807	1107	2C07	4007
08	0800	0008	0808	0804	0802	0801	0867	088B	08CE	08FD	085D	0408	0208	0108	6708	8B08	CE08	FD08	5D08
09	0900	0009	0909	09BC	095E	092F	09A1	09E8	0933	0974	09C0	BC09	5E09	2F09	A109	E809	3309	7409	C009
0A	0A00	000A	0A0A	0A05	0ABA	0A5D	0A9A	0A4D	0A45	0A9E	0A16	050A	BA0A	5D0A	9A0A	4D0A	450A	9E0A	160A
0B	0B00	000B	0B0B	0BBD	0BE6	0B73	0B5C	0B2E	0BB8	0B17	0B8B	BD0B	E60B	730B	5C0B	2E0B	B80B	170B	8B0B
0C	0C00	000C	0C0C	0C06	0C03	0CB9	0CEC	0C76	0CA9	0C3B	0CCB	060C	030C	B90C	EC0C	760C	A90C	3B0C	CBC0
0D	0D00	000D	0D0D	0DBE	0D5F	0D97	0D2A	0D15	0D54	0DB2	0D56	BE0D	5F0D	970D	2A0D	150D	540D	B20D	560D
0E	0E00	000E	0E0E	0E07	0EBB	0EE5	0E11	0EB0	0E22	0E58	0E80	070E	BB0E	E50E	110E	B00E	220E	580E	800E
0F	0F00	000F	0F0F	0FBF	0FE7	0FCB	0FD7	0FD3	0FDF	0FD1	0F1D	BF0F	E70F	CB0F	D70F	D30F	DF0F	D10F	1D0F
10	1000	0010	1010	1008	1004	1002	10CE	1067	10ED	108B	10BA	0810	0410	0210	CE10	6710	ED10	8B10	BA10
11	1100	0011	1111	11B0	1158	112C	1108	1104	1110	1102	1127	B011	5811	2C11	0811	0411	1011	0211	2711
12	1200	0012	1212	1209	12BC	125E	1233	12A1	1266	12E8	12F1	0912	BC12	5E12	3312	A112	6612	E812	F112
13	1300	0013	1313	13B1	13E0	1370	13F5	13C2	139B	1361	136C	B113	E013	7013	F513	C213	9B13	6113	6C13
14	1400	0014	1414	140A	1405	14BA	1445	149A	148A	144D	142C	0A14	0514	BA14	4514	9A14	8A14	4D14	2C14
15	1500	0015	1515	15B2	1559	1594	1583	15F9	1577	15C4	15B1	B215	5915	9415	8315	F915	7715	C415	B115
16	1600	0016	1616	160B	16BD	16E6	16B8	165C	1601	162E	1667	0B16	BD16	E616	B816	5C16	0116	2E16	6716
17	1700	0017	1717	17B3	17E1	17C8	177E	173F	17FC	17A7	17FA	B317	E117	C817	7E17	3F17	FC17	A717	FA17
18	1800	0018	1818	180C	1806	1803	18A9	18EC	1823	1876	18E7	0C18	0618	0318	A918	EC18	2318	7618	E718



**Table 78: x8 ECC correctable syndromes**

Error Bit-mask	Symbol In Error																		
	12h	11h	10h	Fh	Eh	Dh	Ch	Bh	Ah	9h	8h	7h	6h	5h	4h	3h	2h	1h	0h
19	1900	0019	1919	19B4	195A	192D	196F	198F	19DE	19FF	197A	B419	5A19	2D19	6F19	8F19	DE19	FF19	7A19
1A	1A00	001A	1A1A	1A0D	1ABE	1A5F	1A54	1A2A	1AA8	1A15	1AAC	0D1A	BE1A	5F1A	541A	2A1A	A81A	151A	AC1A
1B	1B00	001B	1B1B	1BB5	1BE2	1B71	1B92	1B49	1B55	1B9C	1B31	B51B	E21B	711B	921B	491B	551B	9C1B	311B
1C	1C00	001C	1C1C	1C0E	1C07	1CBB	1C22	1C11	1C44	1CB0	1C71	0E1C	071C	BB1C	221C	111C	441C	B01C	711C
1D	1D00	001D	1D1D	1DB6	1D5B	1D95	1DE4	1D72	1DB9	1D39	1DEC	B61D	5B1D	951D	E41D	721D	B91D	391D	EC1D
1E	1E00	001E	1E1E	1E0F	1EBF	1EE7	1EDF	1ED7	1ECF	1ED3	1E3A	0F1E	BF1E	E71E	DF1E	7D1E	CF1E	D31E	3A1E
1F	1F00	001F	1F1F	1FB7	1FE3	1FC9	1F19	1FB4	1F32	1F5A	1FA7	B71F	E31F	C91F	191F	B41F	321F	5A1F	A71F
20	2000	0020	2020	2010	2008	2004	20ED	20CE	20AB	2067	2005	1020	0820	0420	ED20	CE20	AB20	6720	0520
21	2100	0021	2121	21A8	2154	212A	212B	21AD	2156	21EE	2198	A821	5421	2A21	2B21	AD21	5621	EE21	9821
22	2200	0022	2222	2211	22B0	2258	2210	2208	2220	2204	224E	1122	B022	5822	1022	0822	2022	0422	4E22
23	2300	0023	2323	23A9	23EC	2376	23D6	236B	23DD	238D	23D3	A923	EC23	7623	D623	6B23	DD23	8D23	D323
24	2400	0024	2424	2412	2409	24BC	2466	2433	24CC	24A1	2493	1224	0924	BC24	6624	3324	CC24	A124	9324
25	2500	0025	2525	25AA	2555	2592	25A0	2550	2531	2528	250E	AA25	5525	9225	A025	5025	3125	2825	0E25
26	2600	0026	2626	2613	26B1	26E0	269B	26F5	2647	26C2	26D8	1326	B126	E026	9B26	F526	4726	C226	DB26
27	2700	0027	2727	27AB	27ED	27CE	275D	2796	27BA	274B	2745	AB27	ED27	CE27	5D27	9627	BA27	4B27	4527
28	2800	0028	2828	2814	280A	2805	288A	2845	2865	289A	2858	1428	0A28	0528	8A28	4528	6528	9A28	5828
29	2900	0029	2929	29AC	2956	292B	294C	2926	2998	2913	29C5	AC29	5629	2B29	4C29	2629	9829	1329	C529
2A	2A00	002A	2A2A	2A15	2AB2	2A59	2A77	2A83	2AEE	2AF9	2A13	152A	B22A	592A	772A	832A	EE2A	F92A	132A
2B	2B00	002B	2B2B	2BAD	2BEE	2B77	2BB1	2BE0	2B13	2B70	2B8E	AD2B	EE2B	772B	B12B	E02B	132B	702B	8E2B
2C	2C00	002C	2C2C	2C16	2C0B	2CBD	2C01	2CB8	2C02	2C5C	2CCE	162C	0B2C	BD2C	012C	B82C	022C	5C2C	CE2C
2D	2D00	002D	2D2D	2DAE	2D57	2D93	2DC7	2DDB	2DFF	2DD5	2D53	AE2D	572D	932D	C72D	DB2D	FF2D	D52D	532D
2E	2E00	002E	2E2E	2E17	2EB3	2EE1	2EFC	2E7E	2E89	2E3F	2E85	172E	B32E	E12E	FC2E	7E2E	892E	3F2E	852E
2F	2F00	002F	2F2F	2FAF	2FEF	2FCF	2F3A	2F1D	2F74	2FB6	2F18	AF2F	EF2F	CF2F	3A2F	1D2F	742F	B62F	182F
30	3000	0030	3030	3018	300C	3006	3023	30A9	3046	30EC	30BF	1830	0C30	0630	2330	A930	4630	EC30	BF30
31	3100	0031	3131	31A0	3150	3128	31E5	31CA	31B8	3165	3122	A031	5031	2831	E531	CA31	BB31	6531	2231
32	3200	0032	3232	3219	32B4	325A	32DE	326F	32CD	328F	32F4	1932	B432	5A32	DE32	6F32	3232	8F32	F432
33	3300	0033	3333	33A1	33E8	3374	3318	330C	3330	3306	3369	A133	E833	7433	1833	0C33	3033	0633	6933
34	3400	0034	3434	341A	340D	34BE	34A8	3454	3421	342A	3429	1A34	0D34	BE34	A834	5434	2134	2A34	2934
35	3500	0035	3535	35A2	3551	3590	356E	3537	35DC	35A3	35B4	A235	5135	9035	6E35	3735	DC35	A335	B435
36	3600	0036	3636	361B	36B5	36E2	3655	3692	36AA	3649	3662	1B36	B536	E236	5536	9236	AA36	4936	6236
37	3700	0037	3737	37A3	37E9	37CC	3793	37F1	3757	37C0	37FF	A337	E937	CC37	9337	F137	5737	C037	FF37
38	3800	0038	3838	381C	380E	3807	3844	3822	3888	3811	38E2	1C38	0E38	0738	4438	2238	8838	1138	E238
39	3900	0039	3939	39A4	3952	3929	3982	3941	3975	3998	397F	A439	5239	2939	8239	4139	7539	9839	7F39
3A	3A00	003A	3A3A	3A1D	3AB6	3A5B	3AB9	3AE4	3A03	3A72	3AA9	1D3A	B63A	5B3A	B93A	4E3A	033A	723A	A93A
3B	3B00	003B	3B3B	3BA5	3BEA	3B75	3B7F	3B87	3BFE	3BFB	3B34	A53B	EA3B	753B	7F3B	873B	FE3B	FB3B	343B
3C	3C00	003C	3C3C	3C1E	3C0F	3CBF	3CCF	3CDF	3CEF	3CD7	3C74	1E3C	0F3C	BF3C	CF3C	DF3C	EF3C	D73C	743C
3D	3D00	003D	3D3D	3DA6	3D53	3D91	3D09	3DBC	3D12	3D5E	3DE9	A63D	533D	913D	093D	BC3D	123D	5E3D	E93D
3E	3E00	003E	3E3E	3E1F	3EB7	3EE3	3E32	3E19	3E64	3EB4	3EF3	1F3E	B73E	E33E	323E	193E	643E	B43E	3F3E
3F	3F00	003F	3F3F	3FA7	3FEB	3FCD	3FF4	3F7A	3F99	3F3D	3FA2	A73F	EB3F	CD3F	F43F	7A3F	993F	3D3F	A23F
40	4000	0040	4040	4020	4010	4008	40AB	40ED	4027	40CE	400A	2040	1040	0840	AB40	ED40	2740	CE40	0A40
41	4100	0041	4141	4198	414C	4126	416D	418E	41DA	4147	4197	9841	4C41	2641	6D41	8E41	DA41	4741	9741
42	4200	0042	4242	4221	42A8	4254	4256	422B	42AC	42AD	4241	2142	A842	5442	5642	2B42	AC42	AD42	4142
43	4300	0043	4343	4399	43F4	437A	4390	4348	4351	4324	43DC	9943	F443	7A43	9043	4843	5143	2443	DC43
44	4400	0044	4444	4422	4411	44B0	4420	4410	4440	4408	449C	2244	1144	B044	2044	1044	4044	0844	9C44
45	4500	0045	4545	459A	454D	459E	45E6	4573	45BD	4581	4501	9A45	4D45	9E45	E645	7345	BD45	8145	0145
46	4600	0046	4646	4623	46A9	46EC	46DD	46D6	46CB	466B	46D7	2346	A946	EC46	DD46	D646	CB46	6B46	D746
47	4700	0047	4747	479B	47F5	47C2	471B	47B5	4736	47E2	474A	9B47	F547	C247	1B47	B547	3647	E247	4A47
48	4800	0048	4848	4824	4812	4809	48CC	4866	48E9	4833	4857	2448	1248	0948	CC48	6648	E948	3348	5748
49	4900	0049	4949	499C	494E	4927	490A	4905	4914	49BA	49CA	9C49	E449	2749	0A49	0549	1449	BA49	CA49
4A	4A00	004A	4A4A	4A25	4AAA	4A55	4A31	4AA0	4A62	4A50	4A1C	254A	AA4A	554A	314A	A04A	624A	504A	1C4A
4B	4B00	004B	4B4B	4B9D	4BF6	4B7B	4BF7	4BC3	4B9F	4BD9	4B81	9D4B	F64B	7B4B	F74B	C34B	9F4B	D94B	814B
4C	4C00	004C	4C4C	4C26	4C13	4CB1	4C47	4C9B	4C8E	4CF5	4CC1	264C	134C	B14C	474C	9B4C	8E4C	F54C	C14C
4D	4D00	004D	4D4D	4D9E	4D4F	4D9F	4D81	4DF8	4D73	4D7C	4D5C	9E4D	4F4D	9F4D	814D	F84D	734D	7C4D	5C4D
4E	4E00	004E	4E4E	4E27	4EAB	4EED	4EBA	4E5D	4E05	4E96	4E8A	274E	AB4E	ED4E	BA4E	5D4E	054E	964E	8A4E
4F	4F00	004F	4F4F	4F9F	4FF7	4FC3	4F7C	4F3E	4FF8	4F1F	4F17	9F4F	F74F	C34F	7C4F	3E4F	F84F	1F4F	174F
50	5000	0050	5050	5028	5014	500A	5065	508A	50CA	5045	50B0	2850	1450	0A50	6550	8A50	CA50	4550	B050
51	5100	0051	5151	5190	5148	5124	51A3	51E9	5137	51CC	512D	9051	4851	2451	A351	E951	3751	CC51	2D51
52	5200	0052	5252	5229	52AC	5256	5298	524C	5241	5226	52FB	2952	AC52	5652	9852	4C52	4152	2652	FB52
53	5300	0053	5353	5391	53F0	5378	535E	532F	53BC	53AF	5366	9153	F053	7853	5E53	2F53	BC53	AF53	6653
54	5400	0054	5454	542A	5415	54B2	54EE	5477	54AD	5483	5426	2A54	1554	B254	EE54	7754	AD54	8354	2654
55	5500	0055	5555	5592	5549	559C	5528	5514	5550	550A	55BB	9255	4955	9C55	2855	1455	5055	0A55	BB55
56	5600	0056	5656	562B	56AD	56EE	5613	56B1	5626	56E0	566D	2B56	AD56	EE56	1356	B156	2656	E056	6D56
57	5700	0057	5757	5793	57F1	57C0	57D5	57D2	57DB	5769	57F0	9357	F157	C057	D557	D257	DB57	6957	F057
58	5800	0058	5858	582C	5816	580B	5802	5801	5804	58B8	58ED	2C58	1658	0B58	0258	0158	0458	B858	ED58
59	5900	0059	5959	5994	59A4	5925	59C4	5962	59F9	5931	5970	9459	4A59	2559	C459	6259	F959	3159	7059
5A	5A00	005A	5A5A	5A2D	5AAE	5A57	5AFF	5AC7	5A8F	5ADB	5AA6	2D5A	AE5A	575A	FF5A	C75A	8F5A	DB5A	A65A
5B	5B00	005B	5B5B	5B95	5BF2	5B79	5B39	5BA4	5B72	5B52	5B3B	955B	F25B	795B	395B	A45B	725B	525B	3B5B

**Table 78: x8 ECC correctable syndromes**

Error Bit-mask	Symbol In Error																		
	12h	11h	10h	Fh	Eh	Dh	Ch	Bh	Ah	9h	8h	7h	6h	5h	4h	3h	2h	1h	0h
5C	5C00	005C	5C5C	5C2E	5C17	5CB3	5C89	5CFC	5C63	5C7E	5C7B	2E5C	175C	B35C	895C	FC5C	635C	7E5C	7B5C
5D	5D00	005D	5D5D	5D96	5D4B	5D9D	5D4F	5D9F	5D9E	5DF7	5DE6	965D	4B5D	9D5D	4F5D	9F5D	9E5D	F75D	E65D
5E	5E00	005E	5E5E	5E2F	5EAF	5EEF	5E74	5E3A	5EE8	5E1D	5E30	2F5E	AF5E	EF5E	745E	3A5E	E85E	1D5E	305E
5F	5F00	005F	5F5F	5F97	5FF3	5FC1	5FB2	5F59	5F15	5F94	5FAD	975F	F35F	C15F	B25F	595F	155F	945F	AD5F
60	6000	0060	6060	6030	6018	600C	6046	6023	608C	60A9	600F	3060	1860	0C60	4660	2360	8C60	A960	0F60
61	6100	0061	6161	6188	6144	6122	6180	6140	6171	6120	6192	8861	4461	2261	8061	4061	7161	2061	9261
62	6200	0062	6262	6231	62A0	6250	62BB	62E5	6207	62CA	6244	3162	A062	5062	BB62	E562	0762	CA62	4462
63	6300	0063	6363	6389	63FC	637E	637D	6386	63FA	6343	63D9	8963	FC63	7E63	7D63	8663	FA63	4363	D963
64	6400	0064	6464	6432	6419	64B4	64CD	64DE	64EB	646F	6499	3264	1964	B464	CD64	DE64	EB64	6F64	9964
65	6500	0065	6565	658A	6545	659A	650B	65BD	6516	65E6	6504	8A65	4565	9A65	0B65	BD65	I665	E665	0465
66	6600	0066	6666	6633	66A1	66E8	6630	6618	6660	660C	66D2	3366	A166	E866	3066	1866	6066	0C66	D266
67	6700	0067	6767	678B	67FD	67C6	67F6	677B	679D	6785	674F	8B67	FD67	C667	F667	7B67	9D67	8567	4F67
68	6800	0068	6868	6834	681A	680D	6821	68A8	6842	6854	6852	3468	1A68	0D68	2168	A868	4268	5468	5268
69	6900	0069	6969	698C	6946	6923	69E7	69CB	69BF	69DD	69CF	8C69	4669	2369	E769	BF69	DF69	CF69	
6A	6A00	006A	6A6A	6A35	6AA2	6A51	6ADC	6A6E	6AC9	6A37	6A19	356A	A26A	516A	DC6A	6E6A	C96A	376A	196A
6B	6B00	006B	6B6B	6B8D	6BFE	6B7F	6B1A	6B0D	6B34	6BBE	6B84	8D6B	FE6B	7F6B	1A6B	0D6B	346B	BE6B	846B
6C	6C00	006C	6C6C	6C36	6C1B	6CB5	6CAA	6C55	6C25	6C92	6CC4	366C	1B6C	B56C	AA6C	556C	256C	926C	C46C
6D	6D00	006D	6D6D	6D8E	6D47	6D9B	6D06	6D36	6DD8	6D1B	6D59	8E6D	476D	9B6D	6C6D	366D	I66D	1B6D	596D
6E	6E00	006E	6E6E	6E37	6EA3	6EE9	6E57	6E93	6EAE	6EF1	6E8F	376E	A36E	E96E	576E	936E	AE6E	F16E	8F6E
6F	6F00	006F	6F6F	6F8F	6FFF	6FC7	6F91	6FF0	6F53	6F78	6F12	8F6F	FF6F	C76F	916F	F06F	536F	786F	126F
70	7000	0070	7070	7038	701C	700E	7088	7044	7061	7022	70B5	3870	1C70	0E70	8870	4470	6170	2270	B570
71	7100	0071	7171	7180	7140	7120	714E	7127	719C	71AB	7128	8071	4071	2071	4E71	2771	9C71	AB71	2871
72	7200	0072	7272	7239	72A4	7252	7275	7282	72EA	7241	72FE	3972	A472	5272	7572	8272	EA72	4172	FE72
73	7300	0073	7373	7381	73F8	737C	73B3	73E1	7317	73C8	7363	8173	F873	7C73	B373	E173	1773	C873	6373
74	7400	0074	7474	743A	741D	74B6	7403	74B9	7406	74E4	7423	3A74	1D74	B674	0374	B974	0674	E474	2374
75	7500	0075	7575	7582	7541	7598	75C5	75DA	75FB	756D	75BE	8275	4175	9875	C575	DA75	FB75	6D75	BE75
76	7600	0076	7676	763B	76A5	76EA	76FE	767F	768D	7687	7668	3B76	A576	EA76	FE76	7F76	8D76	8776	6876
77	7700	0077	7777	7783	77F9	77C4	7738	771C	7770	77F5	8377	F977	C477	3877	1C77	7077	0E77	F577	
78	7800	0078	7878	783C	781E	780F	78EF	78CF	78AF	78DF	78E8	3C78	1E78	0F78	EF78	CF78	AF78	DF78	E878
79	7900	0079	7979	7984	7942	7921	7929	79AC	7952	7956	7975	8479	4279	2179	2979	AC79	5279	5679	7579
7A	7A00	007A	7A7A	7A3D	7AA6	7A53	7A12	7A09	7A24	7ABC	7AA3	3D7A	A67A	537A	127A	097A	247A	BC7A	A37A
7B	7B00	007B	7B7B	7B85	7BFA	7B7D	7BD4	7B6A	7BD9	7B35	7B3E	857B	FA7B	7D7B	D47B	6A7B	D97B	357B	7E7B
7C	7C00	007C	7C7C	7C3E	7C1F	7CB7	7C64	7C32	7CC8	7C19	7C7E	3E7C	1F7C	B77C	647C	327C	C87C	197C	7E7C
7D	7D00	007D	7D7D	7D86	7D43	7D99	7DA2	7D51	7D35	7D90	7DE3	867D	437D	997D	A27D	517D	357D	907D	E37D
7E	7E00	007E	7E7E	7E3F	7EA7	7EEB	7E99	7EF4	7E43	7E7A	7E35	3F7E	A77E	EB7E	997E	F47E	437E	7A7E	357E
7F	7F00	007F	7F7F	7F87	7FFB	7FC5	7F5F	7F97	7FBE	7FF3	7FA8	877F	FB7F	C57F	5F7F	977F	BE7F	F37F	A87F
80	8000	0080	8080	8040	8020	8010	8027	80AB	804E	80ED	8014	4080	2080	1080	2780	AB80	4E80	ED80	1480
81	8100	0081	8181	81F8	817C	813E	81E1	81C8	81B3	8164	8189	F881	7C81	3E81	E181	C881	B381	6481	8981
82	8200	0082	8282	8241	8298	824C	82DA	826D	82C5	828E	825F	4182	9882	4C82	DA82	6D82	C582	8E82	5F82
83	8300	0083	8383	83F9	83C4	8362	831C	830E	8338	8307	83C2	F983	C483	6283	1C83	0E83	3883	0783	C283
84	8400	0084	8484	8442	8421	84A8	84AC	8456	8429	842B	8482	4284	2184	A884	AC84	5684	2984	2B84	8284
85	8500	0085	8585	85FA	857D	8586	856A	8535	85D4	85A2	851F	FA85	7D85	8685	6A85	3585	D485	A285	1F85
86	8600	0086	8686	8643	8699	86F4	8651	8690	86A2	8648	86C9	4386	9986	F486	5186	9086	A286	4886	C986
87	8700	0087	8787	87FB	87C5	87DA	8797	87F3	875F	87C1	8754	FB87	C587	DA87	9787	F387	5F87	C187	5487
88	8800	0088	8888	8844	8822	8811	8840	8820	8880	8810	8849	4488	2288	1188	4088	2088	8088	1088	4988
89	8900	0089	8989	89FC	897E	893F	8986	8943	897D	8999	89D4	FC89	7E89	3F89	8689	4389	7D89	9989	D489
8A	8A00	008A	8A8A	8A45	8A9A	8A4D	8ABD	8AE6	8A0B	8A73	8A02	458A	9A8A	4D8A	BD8A	E68A	0B8A	738A	028A
8B	8B00	008B	8B8B	8BFD	8BC6	8B63	8B7B	8B85	8BF6	8BFA	8B9F	FD8B	C68B	638B	7B8B	858B	F68B	FA8B	9F8B
8C	8C00	008C	8C8C	8C46	8C23	8CA9	8CCB	8CDD	8CE7	8CD6	8CDF	468C	238C	A98C	CB8C	DD8C	E78C	D68C	DF8C
8D	8D00	008D	8D8D	8DFE	8D7F	8D87	8D0D	8DBE	8D1A	8D5F	8D42	FE8D	7F8D	878D	0D8D	BE8D	1A8D	5F8D	428D
8E	8E00	008E	8E8E	8E47	8E9B	8EF5	8E36	8E1B	8E6C	8EB5	8E94	478E	9B8E	F58E	368E	1B8E	6C8E	B58E	948E
8F	8F00	008F	8F8F	8FFF	8FC7	8FDB	8FF0	8F78	8F91	8F3C	8F09	FF8F	C78F	DB8F	F08F	788F	918F	3C8F	098F
90	9000	0090	9090	9048	9024	9012	90E9	90CC	90A3	9066	90AE	4890	2490	1290	E990	CC90	A390	6690	AE90
91	9100	0091	9191	91F0	9178	913C	912F	91AF	915E	91EF	9133	F091	7891	3C91	2F91	AC91	5E91	FE91	3391
92	9200	0092	9292	9249	929C	924E	9214	920A	9228	9205	92E5	4992	9C92	4E92	1492	0A92	2892	0592	E592
93	9300	0093	9393	93F1	93C0	9360	93D2	9369	93D5	938C	9378	F193	C093	6093	D293	6993	D593	8C93	7893
94	9400	0094	9494	944A	9425	94AA	9462	9431	94C4	94A0	9438	4A94	2594	AA94	6294	3194	C494	A094	3894
95	9500	0095	9595	95F2	9579	9584	95A4	9552	9539	9529	95A5	F295	7995	8495	A495	5295	3995	2995	A595
96	9600	0096	9696	964B	969D	96F6	969F	96F7	964F	96C3	9673	4B96	9D96	F696	9F96	F796	4F96	C396	7396
97	9700	0097	9797	97F3	97C1	97D8	9759	9794	97B2	974A	97EE	F397	C197	D897	5997	9497	B297	4A97	EE97
98	9800	0098	9898	984C	9826	9813	988E	9847	986D	989B	98F3	4C98	2698	1398	8E98	4798	6D98	9B98	F398
99	9900	0099	9999	99F4	997A	993D	9948	9924	9990	9912	99E6	F499	7A99	3D99	4899	2499	9099	1299	6E99
9A	9A00	009A	9A9A	9A4D	9A9E	9A4F	9A73	9A81	9AE6	9AF8	9AB8	4D9A	9E9A	4F9A	739A	819A	E69A	F89A	B89A
9B	9B00	009B	9B9B	9BF5	9BC2	9B61	9BB5	9BE2	9B1B	9B71	9B25	F59B	C29B	619B	B59B	E29B	1B9B	719B	259B
9C	9C00	009C	9C9C	9C4E	9C27	9CAB	9C05	9CBA	9C0A	9C5D	9C65	4E9C	279C	AB9C	059C	BA9C	0A9C	5D9C	659C
9D	9D00	009D	9D9D	9DF6	9D7B	9D85	9DC3	9DD9	9DF7	9DD4	9DF8	F69D	7B9D	859D	C39D	D99D	F79D	D49D	F89D
9E	9E00	009E	9E9E	9E4F	9E9F	9EF7	9EF8	9E7C	9E81	9E3E	9E2E	4F9E	9F9E	F79E	F89E	7C9E	819E	3E9E	2E9E

Table 78: x8 ECC correctable syndromes

Error Bit-mask	Symbol In Error																		
	12h	11h	10h	Fh	Eh	Dh	Ch	Bh	Ah	9h	8h	7h	6h	5h	4h	3h	2h	1h	0h
9F	9F00	009F	9F9F	9FF7	9FC3	9FD9	9F3E	9F1F	9F7C	9FB7	9FB3	F79F	C39F	D99F	3E9F	1F9F	7C9F	B79F	B39F
A0	A000	00A0	A0A0	A050	A028	A014	A0CA	A065	A0E5	A08A	A011	50A0	28A0	14A0	CAA0	65A0	E5A0	8AA0	11A0
A1	A100	00A1	A1A1	A1E8	A174	A13A	A10C	A106	A118	A103	A18C	E8A1	74A1	3AA1	0CA1	06A1	18A1	03A1	8CA1
A2	A200	00A2	A2A2	A251	A290	A248	A237	A2A3	A2E6	A2E9	A25A	51A2	90A2	48A2	37A2	A3A2	6EA2	E9A2	5AA2
A3	A300	00A3	A3A3	A3E9	A3CC	A366	A3F1	A3C0	A393	A360	A3C7	E9A3	CCA3	66A3	F1A3	C0A3	93A3	60A3	C7A3
A4	A400	00A4	A4A4	A452	A429	A44C	A441	A498	A482	A44C	A487	52A4	29A4	ACA4	41A4	98A4	82A4	4CA4	87A4
A5	A500	00A5	A5A5	A5EA	A575	A582	A587	A5FB	A57F	A5C5	A51A	EAA5	75A5	82A5	87A5	FBA5	7FA5	C5A5	1AA5
A6	A600	00A6	A6A6	A653	A691	A6F0	A6BC	A65E	A609	A62F	A6CC	53A6	91A6	F0A6	BCA6	5EA6	09A6	2FA6	CCA6
A7	A700	00A7	A7A7	A7EB	A7CD	A7DE	A77A	A7D7	A7F4	A7A6	A751	EBA7	CDA7	DEA7	7AA7	3DA7	F4A7	A6A7	51A7
A8	A800	00A8	A8A8	A854	A82A	A815	A8AD	A8EE	A82B	A877	54A8	2AA8	15A8	ADA8	EEA8	02A8	2BA8	77A8	4CA8
A9	A900	00A9	A9A9	A9EC	A976	A93B	A96B	A98D	A9D6	A9FE	A9D1	ECA9	76A9	3BA9	6BA9	8DA9	D6A9	FEA9	D1A9
AA	AA00	00AA	AAAA	AA55	AA92	AA49	AA50	AA28	AAA0	AA14	AA07	55AA	92AA	49AA	50AA	28AA	A0AA	14AA	07AA
AB	AB00	00AB	ABAB	ABED	ABCE	AB67	AB9E	AB4B	AB5D	AB9D	AB9A	EDAB	CEAB	67AB	96AB	4BAB	5DAB	9DAB	9AAB
AC	AC00	00AC	ACAC	AC56	AC2B	ACAD	AC26	AC13	AC4C	ACB1	ACBA	56AC	2BAC	ADAC	26AC	13AC	4CAC	B1AC	DBAC
AD	AD00	00AD	ADAD	ADEE	AD77	AD83	ADE0	AD70	ADB1	AD38	AD47	EEAD	77AD	83AD	E0AD	70AD	B1AD	38AD	47AD
AE	AE00	00AE	AEAE	AE57	AE93	AEF1	AEDB	AED5	AEC7	AED2	AE91	57AE	93AE	F1AE	DBAE	D5AE	C7AE	D2AE	91AE
AF	AF00	00AF	AFAF	AFEF	AFCF	AFFD	AF1D	AFB6	AF3A	AF5B	AF0C	EFAF	CFAF	DFAF	1DAF	B6AF	3AAF	5BAF	0CAF
B0	B000	00B0	B0B0	B058	B02C	B016	B004	B002	B008	B001	B0A8	58B0	2CB0	16B0	04B0	02B0	08B0	01B0	ABB0
B1	B100	00B1	B1B1	B1E0	B170	B138	B1C2	B161	B1F5	B188	B136	E0B1	70B1	38B1	C2B1	61B1	F5B1	88B1	36B1
B2	B200	00B2	B2B2	B259	B294	B24A	B2F9	B2C4	B283	B262	B2E0	59B2	94B2	4AB2	F9B2	C4B2	83B2	62B2	E0B2
B3	B300	00B3	B3B3	B3E1	B3C8	B364	B33F	B3A7	B37E	B3EB	B37D	E1B3	C8B3	64B3	3FB3	A7B3	7EB3	EBB3	7DB3
B4	B400	00B4	B4B4	B45A	B42D	B4AE	B48F	B4FF	B46F	B4C7	B43D	5AB4	2DB4	AEB4	8FB4	FFB4	6FB4	C7B4	3DB4
B5	B500	00B5	B5B5	B5E2	B571	B580	B549	B59C	B592	B54E	B5A0	E2B5	71B5	80B5	49B5	9CB5	92B5	4EB5	A0B5
B6	B600	00B6	B6B6	B65B	B695	B6F2	B672	B639	B6E4	B6A4	B676	5BB6	95B6	F2B6	72B6	39B6	E4B6	A4B6	76B6
B7	B700	00B7	B7B7	B7E3	B7C9	B7DC	B7B4	B75A	B719	B72D	B7EB	E3B7	C9B7	DCB7	B4B7	5AB7	19B7	2DB7	EBB7
B8	B800	00B8	B8B8	B85C	B82E	B817	B863	B889	B8C6	B8FC	B8F6	5CB8	2EB8	17B8	63B8	89B8	C6B8	FCB8	F6B8
B9	B900	00B9	B9B9	B9E4	B972	B939	B9A5	B9EA	B93B	B975	B96B	E4B9	72B9	39B9	A5B9	EAB9	3BB9	75B9	6BB9
BA	BA00	00BA	BABA	BA5D	BA96	BA4B	BA9E	BA4F	BA4D	BA9F	BABD	5DBA	96BA	4BBA	9EBA	4FBA	4DBA	9FBA	DBBA
BB	BB00	00BB	BBBB	BBE5	BBCA	BB65	BB58	BB2C	BBB0	BB16	BB20	E5BB	CABB	65BB	58BB	2CBB	B0BB	16BB	20BB
BC	BC00	00BC	BCBC	BC5E	BC2F	BCAF	BCE8	BC74	BCA1	BC3A	BC60	5EBC	2FBC	AFBC	E8BC	74BC	A1BC	3ABC	60BC
BD	BD00	00BD	BDBD	BDE6	BD73	BD81	BD2E	BD17	BD5C	BDB3	BDFD	E6BD	73BD	81BD	2EBD	17BD	5CBD	B3BD	FDBD
BE	BE00	00BE	BEBE	BE5F	BE97	BEF3	BE15	BEB2	BE2A	BE59	BE2B	5FBE	97BE	F3BE	15BE	B2BE	2ABE	59BE	2BBE
BF	BF00	00BF	BFBF	BFE7	BFCB	BFFD	BFD3	BFD1	BFD7	BFD0	BFB6	E7BF	CBBF	DDBF	D3BF	D1BF	D7BF	D0BF	B6BF
C0	C000	00C0	C0C0	C060	C018	C008	C046	C0C6	C069	C023	C01E	60C0	30C0	18C0	8CC0	46C0	69C0	23C0	1EC0
C1	C100	00C1	C1C1	C1D8	C16C	C136	C14A	C125	C194	C1AA	C183	D8C1	6CC1	36C1	4AC1	25C1	94C1	AAC1	83C1
C2	C200	00C2	C2C2	C261	C288	C244	C271	C280	C2E2	C240	C255	61C2	88C2	44C2	71C2	80C2	E2C2	40C2	55C2
C3	C300	00C3	C3C3	C3D9	C3D4	C36A	C3B7	C3E3	C31F	C3C9	C3C8	D9C3	D4C3	6AC3	B7C3	E3C3	1FC3	C9C3	C8C3
C4	C400	00C4	C4C4	C462	C431	C4A0	C407	C4BB	C40E	C4E5	C488	62C4	31C4	A0C4	07C4	BBC4	0EC4	E5C4	88C4
C5	C500	00C5	C5C5	C5DA	C56D	C58E	C5C1	C5D8	C5F3	C56C	C515	DAC5	6DC5	8EC5	C1C5	D8C5	F3C5	6CC5	15C5
C6	C600	00C6	C6C6	C663	C689	C6FC	C6FA	C67D	C685	C686	C6C3	63C6	89C6	FCC6	FAC6	7DC6	85C6	86C6	C3C6
C7	C700	00C7	C7C7	C7B6	C7D5	C7D2	C73C	C71E	C778	C70F	C75E	DBC7	D5C7	D2C7	3CC7	1EC7	78C7	0FC7	5EC7
C8	C800	00C8	C8C8	C864	C832	C819	C8EB	C8CD	C8A7	C8DE	C843	64C8	32C8	19C8	EBC8	CDC8	A7C8	DEC8	43C8
C9	C900	00C9	C9C9	C9DC	C96E	C937	C92D	C9AE	C95A	C957	C9DE	DCC9	6EC9	37C9	2DC9	AEC9	5AC9	57C9	DEC9
CA	CA00	00CA	CACA	CA65	CA8A	CA45	CA16	CA0B	CA2C	CABD	CA08	65CA	8ACA	45CA	16CA	0BCA	2CCA	BDCA	08CA
CB	CB00	00CB	CBCB	CBDD	CBDE	CB6B	CBD0	CB68	CBD1	CB34	CB95	DDCB	D6CB	6BCB	D0CB	68CB	D1CB	34CB	95CB
CC	CC00	00CC	CCCC	CC66	CC33	CCA1	CC60	CC30	CC00	CC18	CCD5	66CC	33CC	A1CC	60CC	30CC	00CC	18CC	D5CC
CD	CD00	00CD	CDCD	CDDE	CD6F	CD8F	CDA6	CD53	CD3D	CD91	CD48	DECD	6FCD	8FCD	A6CD	53CD	3DCD	91CD	48CD
CE	CE00	00CE	CECE	CE67	CE8B	CEFD	CE9D	CEF6	CE4B	CE7B	CE9E	67CE	8BCE	FDCE	9DCE	F6CE	4BCE	7BCE	9ECE
CF	CF00	00CF	CFCF	CFDF	CFD7	CFD3	CF5B	CF95	CFB6	CFE7	CF03	DFCF	D7CF	D3CF	5BCF	95CF	B6CF	F2CF	03CF
D0	D000	00D0	D0D0	D068	D034	D01A	D042	D021	D084	D0A8	D0A4	68D0	34D0	1AD0	42D0	21D0	84D0	A8D0	A4D0
D1	D100	00D1	D1D1	D1D0	D168	D134	D184	D142	D179	D121	D139	D0D1	68D1	34D1	84D1	42D1	79D1	21D1	39D1
D2	D200	00D2	D2D2	D269	D28C	D246	D2BF	D2E7	D20F	D2CB	D2EF	69D2	8CD2	46D2	BFD2	E7D2	0FD2	CB2D	EFD2
D3	D300	00D3	D3D3	D3D1	D3D0	D368	D379	D384	D3F2	D342	D372	D1D3	D0D3	68D3	79D3	84D3	F2D3	42D3	72D3
D4	D400	00D4	D4D4	D46A	D435	D4A2	D4C9	D4DC	D4E3	D46E	D432	6AD4	35D4	A2D4	C9D4	DCD4	E3D4	6ED4	32D4
D5	D500	00D5	D5D5	D5D2	D569	D58C	D50F	D5BF	D51E	D5E7	D5AF	D2D5	69D5	8CD5	0FD5	BFD5	1ED5	E7D5	AFD5
D6	D600	00D6	D6D6	D66B	D68D	D6FE	D634	D61A	D668	D6D0	D679	6BD6	8DD6	FED6	34D6	1AD6	68D6	0DD6	79D6
D7	D700	00D7	D7D7	D7D3	D7D1	D7D0	D7F2	D779	D795	D784	D7EA	D3D7	D1D7	D0D7	F2D7	79D7	95D7	84D7	E4D7
D8	D800	00D8	D8D8	D86C	D836	D81B	D825	D8AA	D84A	D855	D8F9	6CD8	36D8	1BD8	25D8	AAD8	4AD8	55D8	F9D8
D9	D900	00D9	D9D9	D9D4	D96A	D935	D9E3	D9C9	D9B7	D9DC	D964	D4D9	6AD9	35D9	E3D9	C9D9	B7D9	DCD9	64D9
DA	DA00	00DA	DADA	DA6D	DA8E	DA47	DAD8	DA6C	DAC1	DA36	DAB2	6DDA	8EDA	47DA	D8DA	6CDA	C1DA	36DA	B2DA
DB	DB00	00DB	DBDB	DBD5	DBD2	DB69	DB1E	DB0F	DB3C	DBBF	DB2F	D5DB	D2DB	69DB	1EDB	0FDB	3CDB	BFDB	2FDB
DC	DC00	00DC	DCDC	DC6E	DC37	DCA3	DCAE	DC57	DC2D	DC93	DC6F	6EDC	37DC	A3DC	AEDC	57DC	2DDC	93DC	6FDC
DD	DD00	00DD	DDDD	DDD6	DD6B	DD8D	DD68	DD34	DDD0	DD1A	DDF2	D6DD	6BDD	8DDD	68DD	34DD	D0DD	1ADD	F2DD
DE	DE00	00DE	DEDE	DE6F	DE8F	DEFF	DE53	DE91	DEA6	DEF0	DE24	6FDE	8FDE	FFDE	53DE	91DE	A6DE	F0DE	24DE
DF	DF00	00DF	DFDF	DFD7	DFD3	DFD1	DF95	DFE2	DF5B	DF79	DFB9	D7DF	D3DF	D1DF	95DF	F2DF	5BDF	79DF	B9DF
E0	E000	00E0	E0E0	E070	E038	E01C	E061	E088	E0C2	E044	E01B	70E0	38E0	1CE0	61E0	88E0	C2E0	44E0	1BE0
E1	E100	00E1	E1E1	E1C8	E164	E132	E1A7	E1EB	E13F	E1CD	E186	C8E1	64E1	32E1	A7E1	E1E1	3FE1	CDE1	86E1

**Table 78: x8 ECC correctable syndromes**

Error Bit-mask	Symbol In Error																			
	12h	11h	10h	Fh	Eh	Dh	Ch	Bh	Ah	9h	8h	7h	6h	5h	4h	3h	2h	1h	0h	
E2	E200	00E2	E2E2	E271	E280	E240	E29C	E24E	E249	E227	E250	71E2	80E2	40E2	9CE2	4EE2	49E2	27E2	50E2	
E3	E300	00E3	E3E3	E3C9	E3DC	E36E	E35A	E32D	E3B4	E3AE	E3CD	C9E3	DCE3	6EE3	5AE3	2DE3	B4E3	AEE3	CDE3	
E4	E400	00E4	E4E4	E472	E439	E4A4	E4EA	E475	E4A5	E482	E48D	72E4	39E4	A4E4	EAE4	75E4	A5E4	82E4	8DE4	
E5	E500	00E5	E5E5	E5CA	E565	E58A	E52C	E516	E558	E50B	E510	CAE5	6E5E	8AE5	2CE5	16E5	58E5	0BE5	10E5	
E6	E600	00E6	E6E6	E673	E681	E6F8	E617	E6B3	E62E	E6E1	E6C6	73E6	81E6	F8E6	17E6	B3E6	2EE6	E1E6	C6E6	
E7	E700	00E7	E7E7	E7CB	E7DD	E7D6	E7D1	E7D0	E7D3	E768	E75B	CBE7	DDE7	D6E7	D1E7	D0E7	D3E7	68E7	5BE7	
E8	E800	00E8	E8E8	E874	E83A	E81D	E806	E803	E80C	E8B9	E846	74E8	3AE8	1DE8	06E8	03E8	0CE8	B9E8	46E8	
E9	E900	00E9	E9E9	E9CC	E966	E933	E9C0	E960	E9F1	E930	E9DB	CCE9	66E9	33E9	C0E9	60E9	F1E9	30E9	DBE9	
EA	EA00	00EA	EAEA	EA75	EA82	EA41	EAFB	EAC5	EA87	EADA	EA0D	75EA	82EA	41EA	FBEA	C5EA	87EA	DAEA	0DEA	
EB	EB00	00EB	E3EB	EBCD	EBDE	EB6F	EB3D	EBA6	EB7A	EB5E	EB90	CDEB	DEEB	6FEB	3DEB	7AEB	68EB	90EB		
EC	EC00	00EC	ECEC	EC76	EC3B	ECA5	EC8D	ECFE	EC6B	EC7F	ECD0	76EC	3BEC	A5EC	8DEC	FEEC	6BEC	7FEC	D0EC	
ED	ED00	00ED	EDED	EDCE	ED67	ED8B	ED4B	ED9D	ED96	EDF6	ED4D	CEED	67ED	8BED	4BED	9DED	96ED	F6ED	4DED	
EE	EE00	00EE	EEEE	EE77	EE83	EEF9	EE70	EE38	EEEE	EE1C	EE9B	77EE	83EE	F9EE	70EE	38EE	E0EE	1CEE	9BEE	
EF	EF00	00EF	EFEF	EFCF	EFD7	EFD7	EFB6	EF5B	EF1D	EF95	EF06	CFEF	DFEF	D7EF	B6EF	5BEF	1DEF	95EF	06EF	
F0	F000	00F0	F0F0	F078	F03C	F01E	F0AF	F0EF	F02F	F0CF	F0A1	78F0	3CF0	1EF0	AFF0	EFF0	2FF0	CFF0	A1F0	
F1	F100	00F1	F1F1	F1C0	F160	F130	F169	F18C	F1D2	F146	F13C	C0F1	60F1	30F1	69F1	8CF1	D2F1	46F1	3CF1	
F2	F200	00F2	F2F2	F279	F284	F242	F252	F229	F2A4	F2AC	F2EA	79F2	84F2	42F2	52F2	29F2	A4F2	ACF2	EAF2	
F3	F300	00F3	F3F3	F3C1	F3D8	F36C	F394	F34A	F359	F325	F377	C1F3	D8F3	6CF3	94F3	4AF3	59F3	25F3	77F3	
F4	F400	00F4	F4F4	F47A	F43D	F4A6	F424	F412	F448	F409	F437	7AF4	3DF4	A6F4	24F4	12F4	48F4	09F4	37F4	
F5	F500	00F5	F5F5	F5C2	F561	F588	F5E2	F571	F5B5	F580	F5AA	C2F5	61F5	88F5	E2F5	71F5	B5F5	80F5	AAF5	
F6	F600	00F6	F6F6	F67B	F685	F6FA	F6D9	F6D4	F6C3	F66A	F67C	7BF6	85F6	FAF6	D9F6	D4F6	C3F6	6AF6	7CF6	
F7	F700	00F7	F7F7	F7C3	F7D9	F7D4	F71F	F7B7	F73E	F7E3	F7E1	C3F7	D9F7	D4F7	1FF7	B7F7	3EF7	E3F7	E1F7	
F8	F800	00F8	F8F8	F87C	F83E	F81F	F8C8	F864	F8E1	F832	F8FC	7CF8	3EF8	1FF8	C8F8	64F8	E1F8	32F8	FCF8	
F9	F900	00F9	F9F9	F9C4	F962	F931	F90E	F907	F91C	F9BB	F961	C4F9	62F9	31F9	0EF9	07F9	1CF9	BBF9	61F9	
FA	FA00	00FA	FAFA	FA7D	FA86	FA43	FA35	FAA2	FA6A	FA51	FAB7	7DFA	86FA	43FA	35FA	A2FA	6AFA	51FA	B7FA	
FB	FB00	00FB	FBFB	FBC5	FBDA	FB6D	FBF3	FBC1	FB97	FBD8	FB2A	C5FB	DAFB	6DFB	F3FB	C1FB	97FB	D8FB	2AFB	
FC	FC00	00FC	FCFC	FC7E	FC3F	FCA7	FC43	FC99	FC86	FCF4	FC6A	7EFC	3FFC	A7FC	43FC	99FC	86FC	F4FC	6AFC	
FD	FD00	00FD	FDFD	FDC6	FD63	FD89	FD85	FDFA	FD7B	FD7D	FDF7	C6FD	63FD	89FD	85FD	FAFD	7BFD	7DFD	F7FD	
FE	FE00	00FE	FEFE	FE7F	FE87	FEFB	FEBE	FE5F	FE0D	FE97	FE21	7FFE	87FE	FBFE	BEFE	5FFE	0DFE	97FE	21FE	
FF	FF00	00FF	FFFF	FFC7	FFDB	FFD5	FF78	FF3C	FFF0	FF1E	FFBC	C7FF	DBFF	D5FF	78FF	3CFF	F0FF	1EFF	BCFF	

**2.12.2.3.2 x4 ECC**

The use of x4 ECC is indicated in [F3x180\[EccSymbolSize\]](#).

The syndrome field uniquely identifies the failing bit positions of a correctable ECC error. Only syndromes identified by [Table 79](#) 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 79](#), first find the 16-bit syndrome value in the table. This is most easily done by using low order 4 bits of the syndrome to select the appropriate error bitmask column. The entire four digit syndrome should then be in one of the rows of that column. The Symbol In Error row indicates which symbol, and therefore which DIMM has the error, and the column indicates which bits within the symbol. To map to the DIMM, use the algorithm in section [2.8.6 \[Routing DRAM Requests\]](#).

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 79: 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	bcfd	734e	369f
Data 17	63e1	b172	d293	14b4	7755	a5c6	c627	28d8	4b39	99aa	fa4b	3c6c	5f8d	8d1e	eeff
Data 18	b741	d982	6ec3	2254	9515	fbd6	4c97	33a8	84e9	ea2a	5d6b	11fc	a6bd	c87e	7f3f
Data 19	dd41	6682	bbc3	3554	e815	53d6	8e97	1aa8	c7e9	7c2a	a16b	2ffc	f2bd	497e	943f
Data 20	2bd1	3d62	16b3	4f34	64e5	7256	5987	8518	aec9	b87a	93ab	ca2c	e1fd	f74e	dc9f
Data 21	83c1	c142	4283	a4f4	2735	65b6	e677	f858	7b99	391a	badb	5cac	df6d	9dee	1e2f
Data 22	8fd1	c562	4ab3	a934	26e5	6c56	e387	fe18	71c9	3b7a	b4ab	572c	d8fd	924e	1d9f
Data 23	4791	89e2	ce73	5264	15f5	db86	9c17	a3b8	e429	2a5a	6dcb	f1dc	b64d	783e	3faf
Data 24	5781	a9c2	fe43	92a4	c525	3b66	6ce7	e3f8	b479	4a3a	1dbb	715c	26dd	d89e	8f1f
Data 25	bf41	d582	6ac3	2954	9615	fcd6	4397	3ea8	81e9	eb2a	546b	17fc	a8bd	c27e	7d3f
Data 26	9391	e1e2	7273	6464	f7f5	8586	1617	b8b8	2b29	595a	cacb	dcdc	4f4d	3d3e	aeaf
Data 27	cce1	4472	8893	fdb4	3155	b9c6	7527	56d8	9a39	12aa	de4b	ab6c	678d	ef1e	23ff
Data 28	a761	f9b2	5ed3	e214	4575	1ba6	bcc7	7328	d449	8a9a	2dfb	913c	365d	688e	cfef
Data 29	ff61	55b2	aad3	7914	8675	2ca6	d3c7	9e28	6149	cb9a	34fb	e73c	185d	b28e	4def
Data 30	5451	a8a2	fcf3	9694	c2c5	3e36	6a67	ebe8	bf9b	434a	171b	7d7c	292d	d5de	818f
Data 31	6fc1	b542	da83	19f4	7635	acb6	c377	2e58	4199	9b1a	f4db	37ac	586d	82ee	ed2f
Check0	be01	d702	6903	2104	9f05	f606	4807	3208	8c09	e50a	5b0b	130c	ad0d	c40e	7a0f
Check1	4101	8202	c303	5804	1905	da06	9b07	ac08	ed09	2e0a	6f0b	f40c	b50d	760e	370f
Check2	c441	4882	8cc3	f654	3215	bed6	7a97	5ba8	9fe9	132a	d76b	adfc	69bd	e57e	213f

**Table 79: x4 ECC correctable syndromes**

Symbol In Error	Error Bitmask														
	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Check3	7621	9b32	ed13	da44	ac65	4176	3757	6f88	19a9	f4ba	829b	b5cc	c3ed	2efe	58df

### 2.12.3 Sideband Interface (SBI)

The sideband interface (SBI) is an SMBus v2.0 compatible 2-wire processor slave interface. SBI is also referred as the Advanced Platform Management Link. All I2C v2.1 speeds are supported.

SBI is used to communicate with the Remote Management Interface (SB-RMI) (see the *Advanced Platform Management Link (APML) Specification*) and the Temperature Sensor Interface (SB-TSI). Revision C and earlier is compliant to version 1.0 of SB-TSI. Revision D and later is compliant to version 1.5 of SB-TSI and version 1.0 of APML. See [F3x1EC\\_x\[FF:01\]](#)[SB-TSI Interface Revision Register] and [F3x1EC\\_x1\[7F:00\]](#)[SB-RMI Interface Revision Register].

#### 2.12.3.1 SBI Processor Information

Processor access to the SBI configuration is via [\[The SBI Control Register\] F3x1E4](#). The processor can access SB-TSI and SB-RMI registers through [\[The SBI Address Register\] F3x1E8](#) and [\[The SBI Data Register\] F3x1EC](#).

## 2.13 Interrupts

### 2.13.1 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 [F0x68\[ApicExtBrdCst\]](#), then the interrupt is a broadcast interrupt and is accepted by all local APICs regardless of destination mode.

### 2.13.1.1 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.

### 2.13.1.2 Logical Destination Mode

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

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

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

### 2.13.1.3 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 [\[The Interrupt Request Registers\] APIC\[270:200\]](#) corresponding to the vector in the interrupt. For local interrupt sources, this comes from the vector field in that interrupt's local vector table entry. The corresponding bit in [\[The Trigger Mode Registers\] APIC\[1F0:180\]](#) 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.

### 2.13.1.4 Vectored Interrupt Handling

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

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

The processor priority is used to determine if any accepted interrupts (indicated by [APIC\[270:200\]\[RequestBits\]](#)) are high enough priority to be serviced by the processor. When the processor is ready to service an interrupt, the highest bit in [APIC\[270:200\]\[RequestBits\]](#) is cleared, and the corresponding bit is set in [APIC\[170:100\]\[InServiceBits\]](#).

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



### 2.13.1.5 Interrupt Masking

Interrupt masking is controlled by the [The Extended APIC Control Register] APIC410. If APIC410[IerCap] is set, [The Interrupt Enable Registers] APIC[4F0:480] are used to mask interrupts. Any bit in APIC[4F0:480][InterruptEnableBits] that is clear indicates the corresponding interrupt is masked. A masked interrupt is not serviced and the corresponding bit in APIC[270:200][RequestBits] remains set.

### 2.13.1.6 Spurious Interrupts

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

#### 2.13.1.6.1 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.

### 2.13.1.7 Lowest-Priority Interrupt Arbitration

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

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

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

highest enabled interrupt. The main priority bits 7-4 are compared as follows:

```
If (APIC80[Priority[7:4]] >= IRRVec[7:4] and APIC80[Priority[7:4]] > ISRVect[7:4])
Then APIC90[Priority] = APIC80[Priority]
Else if (IRRVec[7:4] > ISRVect[7:4]) APIC90[Priority] = {IRRVec[7:4],0h}
Else APIC90[Priority] = {ISRVect[7:4],0h}
```

### 2.13.1.8 Inter-Processor Interrupts

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

### 2.13.1.9 APIC Timer Operation

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

### 2.13.1.10 Generalized Local Vector Table

All LVTs (APIC320 through APIC370 and APIC[530:500]) support a generalized message type. The generalized values for MsgType are:

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

### 2.13.1.11 State at Reset

At power-up or reset, the APIC is hardware disabled (MSR0000\_001B[ApicEn]=0) so only SMI, NMI, INIT, and ExtInt interrupts may be accepted.

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

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

## 2.13.2 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.

### 2.13.2.1 SMM Overview

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

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

### 2.13.2.2 Operating Mode and Default Register Values

The software environment after entering SMM has the following characteristics:

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

The SMM base address is specified by [\[The SMM Base Address Register \(SMM\\_BASE\)\] MSRC001\\_0111\[SMM\\_BASE\]](#). Important offsets to the base address pointer are:

- [MSRC001\\_0111\[SMM\\_BASE\]](#) + 8000h: SMI handler entry point.
- [MSRC001\\_0111\[SMM\\_BASE\]](#) + FE00h - FFFFh: SMM state save area.

### 2.13.2.3 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 [\[The SMI Trigger IO Cycle Register\] MSRC001\\_0056](#) are:

- In the core, as specified by:

- [The Machine Check Exception Redirection Register] MSRC001\_0022.
- [The IO Trap Registers (SMI\_ON\_IO\_TRAP\_[3:0])] MSRC001\_00[53:50].
- In the NB, as specified by:
  - [The On-Line Spare Control Register] F3xB0.
  - [The NB Machine Check Misc (Thresholding) Registers] F3x1[78, 70, 68, 60].
- All local APIC LVT registers programmed to generate SMIs.

The status for these is stored in [SMMFEC4](#).

In addition, there are SMI events that trigger IO cycles defined by [The Interrupt Pending and CMP-Halt Register] [MSRC001\\_0055](#); see that register for the events.

#### 2.13.2.4 SMM Initial State

After storing the save state, execution starts at [MSRC001\\_0111](#)[SMM\_BASE] + 08000h. The SMM initial state is specified in the following table.

**Table 80: SMM initial state**

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

#### 2.13.2.5 SMM Save State

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

Table 81: 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	
FE80h	4 Bytes	IDTR	reserved	Read-only
FE84h	Word		Limit	
FE86h	2 Bytes		reserved	
FE88h	Quadword		Base	
FE90h	Word	TR	Selector	Read-only
FE92h	Word		Attributes	
FE94h	Doubleword		Limit	
FE98h	Quadword		Base	

**Table 81: SMM Save State**

Offset	Size	Contents	Access
FEA0h	Quadword	IO_RESTART_RIP	Read-only
FEA8h	Quadword	IO_RESTART_RCX	
FEB0h	Quadword	IO_RESTART_RSI	
FEB8h	Quadword	IO_RESTART_RDI	
FEC0h	Doubleword	[The SMM IO Trap Offset] SMMFEC0	Read-only
FEC4	Doubleword	[The Local SMI Status] SMMFEC4	Read-only
FEC8h	Byte	[The SMM IO Restart Byte] SMMFEC8	Read-write
FEC9h	Byte	[The Auto Halt Restart Offset] SMMFEC9	Read-write
FECAh	Byte	[The NMI Mask] SMMFECA	Read-write
FECBh	5 Bytes	reserved	
FED0h	Quadword	EFER	Read-only
FED8h	Quadword	SVM State	Read-only
FEE0h	Quadword	Guest VMCB physical address	Read-only
FEE8h	Quadword	SVM Virtual Interrupt Control	Read-only
FEF0h	16 Bytes	reserved	
FEFCh	Doubleword	[The SMM-Revision Identifier] SMMFEFC	Read-only
FF00h	Doubleword	[The SMM Base Address Register (SMM_BASE)] SMMFF00	Read-write
FF04h	28 Bytes	reserved	
FF20h	Quadword	Guest PAT	Read-only
FF28h	Quadword	Host EFER <sup>2</sup>	
FF30h	Quadword	Host CR4 <sup>2</sup>	
FF38h	Quadword	Host CR3 <sup>2</sup>	
FF40h	Quadword	Host Cr0 <sup>2</sup>	
FF48h	Quadword	CR4	
FF50h	Quadword	CR3	
FF58h	Quadword	CR0	
FF60h	Quadword	DR7	
FF68h	Quadword	DR6	
FF70h	Quadword	RFLAGS	Read-write
FF78h	Quadword	RIP	Read-write
FF80h	Quadword	R15	
FF88h	Quadword	R14	
FF90h	Quadword	R13	
FF98h	Quadword	R12	
FFA0h	Quadword	R11	
FFA8h	Quadword	R10	
FFB0h	Quadword	R9	
FFB8h	Quadword	R8	

**Table 81: SMM Save State**

Offset	Size	Contents	Access
FFC0h	Quadword	RDI	Read-write
FFC8h	Quadword	RSI	
FFD0h	Quadword	RBP	
FFD8h	Quadword	RSP	
FFE0h	Quadword	RBX	
FFE8h	Quadword	RDX	
FFF0h	Quadword	RCX	
FFF8h	Quadword	RAX	
Notes:			
1. This notation specifies that bit[47] is replicated in each of the 16 MSBs of the DW (sometimes called <i>sign extended</i> ). The 16 LSBs 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, [\[The SMM IO Trap Offset\]](#) SMMFEC0 contains information about that IO instruction. For example, if an IO access targets an unavailable device, the system can assert SMI and trap the IO instruction. SMMFEC0 then provides the SMI handler with information about the IO instruction that caused the trap. After the SMI handler takes the appropriate action, it can reconstruct and then re-execute the IO instruction from SMM. Or, more likely, it can use [\[The SMM IO Restart Byte\]](#) SMMFEC8, to cause the core to re-execute the IO instruction immediately after resuming from SMM.

Bits	Description
31:16	<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
31:23	Reserved.
22	<b>SmiSrcOnLineSpare: SMI source on-line spare.</b> This bit is associated with the SMI sources specified in [The On-Line Spare Control Register] F3xB0.
21	Reserved.
20	<b>SmiSrcThrCntL3: SMI source L3 cache thresholding.</b> This bit is associated with the SMI source specified in the L3 cache thresholding register (see [The NB Machine Check Misc (Thresholding) Registers] F3x1[78, 70, 68, 60]).
19	<b>SmiSrcThrCntHt: SMI source link thresholding.</b> This bit is associated with the SMI source specified in the link thresholding register (see [The NB Machine Check Misc (Thresholding) Registers] F3x1[78, 70, 68, 60]).
18	<b>SmiSrcThrCntDram: SMI source DRAM thresholding.</b> This bit is associated with the SMI source specified in the DRAM thresholding register (see [The NB Machine Check Misc (Thresholding) Registers] F3x1[78, 70, 68, 60]).
17	<b>SmiSrcLvtExt: SMI source LVT extended entry.</b> This bit is associated with the SMI sources specified in [The Extended Interrupt [3:0] Local Vector Table Registers] APIC[530:500].
16	<b>SmiSrcLvtLcy: SMI source LVT legacy entry.</b> This bit is associated with the SMI sources specified by the non-extended LVT entries of the APIC.
15:11	Reserved.
10	<b>IntPendSmiSts: interrupt pending SMI status.</b> This bit is associated with the SMI source specified in [The Interrupt Pending and CMP-Halt Register] MSRC001_0055[IntrPndMsg] (when that bit is high).
9	<b>SmiOnCmpHaltSts: SMI on CMP halt status.</b> This bit is associated with the SMI source specified in [The Interrupt Pending and CMP-Halt Register] MSRC001_0055[SmiOnCmpHalt].
8	<b>MceRedirSts: machine check exception redirection status.</b> This bit is associated with the SMI source specified in [The Machine Check Exception Redirection Register] MSRC001_0022[RedirSmiEn].
7:4	Reserved.
3:0	<b>IoTrapSts: IO trap status.</b> Each of these bits is associated with each of the respective SMI sources specified in [The IO Trap Registers (SMI_ON_IO_TRAP_[3:0])] MSRC001_00[53:50].

### SMMFEC8 SMM IO Restart Byte

00h on entry into SMM.

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

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

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

Bits	Description
7:0	<b>RST: SMM IO Restart Byte.</b> Read-write.

### SMMFEC9 Auto Halt Restart Offset

Bits	Description
7:1	Reserved.
0	<p><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.</p> <p>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.</p>

### 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

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> Revision RB-C2, BL-C2 and earlier: Reserved. Revision DA-C2 and C3 and later: Read-only.												
2:0	<b>SVM State.</b> Read-only. <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> </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.
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.												

### 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 Register (SMM\_BASE)

This offset is loaded with the contents of [MSRC001\\_0111](#). See that register for more details.

#### 2.13.2.6 Exceptions and Interrupts in SMM

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

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

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

#### 2.13.2.7 The Protected ASeg and TSeg Areas

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

#### 2.13.2.8 SMM Special Cycles

Special cycles can be initiated on entry and exit from SMM to acknowledge to the system that these transitions are occurring. These are controlled by [MSRC001\\_0015](#)[SMISPCYCDIS, RSMSPCYCDIS].

### 2.13.2.9 Locking SMM

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

### 2.13.2.10 Multiple Unsynchronized SMI Sources

When more than one IO device in the system is enabled to signal an SMI, or when a single device may signal multiple SMI messages without hardware synchronization (e.g. using an end of SMI gate), the processor cores may enter a state where all cores' SMI interrupt pending status bits do not match. As a result, the application processor cores, which are usually slaved by the boot strap core in handling SMIs and controlling SMM flow, may enter a software loop in SMM.

The BIOS must take special care to ensure that all cores have entered SMM prior to accessing shared IO resources and all processors' SMI interrupt status bits are synchronized. The act of synchronizing cores into SMM is called spring boarding. SMI spring boarding applies to all multi-core processors that are affected by the platform architectural factors mentioned above.

An ACPI-compliant IO hub is required for SMM 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 SMI spring boarding are as follows.

- A binary semaphore located in SMRAM, accessible by all processors. For the purpose of this discussion, the semaphore is called CheckSpringBoard. CheckSpringBoard is initialized to zero.
- Two semaphores located in SMRAM, accessible by all processors. 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 processor 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 processor 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 processor core execution at the end of the task. After all processor cores have executed, proceed to step 4. The following is a brief description of the task for each processor core:
  - Check all enabled processor-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 boot strap processor to handle.
  - An exclusive mailbox must exist for each processor core for each core local event.
  - On-line spare events should be handled in this task by the individual core for optimal performance. Assign one core of a dual core processor to handle On-line spare. These events may be optionally handled by the BSC just as other global events.
  - Wait for all processor cores to complete this task at least once.
4. If the current processor 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 processor cores entering SMM. The maximum SMM entrance latency is defined by the platform's IO sub-system, not the processor. AMD recommends a value of twice the watchdog timer count. See [\[The MCA NB Configuration Register\] F3x44](#) 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.

## 2.14 Secure Virtual Machine Mode (SVM)

Support for SVM mode is indicated by [CPUID Fn8000\\_0001\\_ECX\[SVM\]](#). If SVM is supported, then the DEV registers starting at [F3xF0](#) are visible.

### 2.14.1 BIOS support for SVM Disable

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

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

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

## 2.15 CPUID Instruction

The CPUID instruction provides data about the features supported by the processor. See 3.9 [CPUID Instruction Registers] for details.

### 2.15.1 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].

### 2.15.2 L3 Cache Support

The BIOS must determine if the processor includes a third level memory cache (L3) by reading [The L2/L3 Cache and L2 TLB Identifiers] CPUID Fn8000\_0006 and take steps to correctly display cache size information on the POST video screen:

- Issue CPUID Fn8000\_0006. If EDX[L3Size] is not zero then the processor includes an L3. The L3Size field indicates the L3 cache size.
- If the *total* cache size is displayed on the screen then the BIOS must correctly calculate the total of L1+L2+L3 sizes.
- It is preferred that the BIOS shows the exact breakdown between the L1, L2, and L3 cache sizes and the total. For example, specify L1 (128 Kbytes) + L2 (size of L2 in Kbytes) + L3 (size of L3 in Kbytes) = total cache size in Kbytes.

## 2.16 Performance Monitoring

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

### 2.16.1 Performance Monitor Counters

The performance monitor counters are used by software to count specific events that occur in the processor. [The Performance Event Select Register (PERF\_CTL[3:0])] MSRC001\_00[03:00] and [The Performance Event Counter Registers (PERF\_CTR[3:0])] MSRC001\_00[07:04] specify the events to be monitored and how they are monitored. All of the events are specified in section 3.14 [Performance Counter Events].

### 2.16.2 Instruction Based Sampling (IBS)

IBS is a code profiling mechanism that enables the processor to select a random instruction fetch or micro-op after a programmed time interval has expired and record specific performance information about the operation. An interrupt is generated when the operation is complete as specified by [The IBS Control Register] F3x1CC.

An interrupt handler can then read the performance information that was logged for the operation.

The IBS mechanism is split into two parts: instruction fetch performance controlled through [The IBS Fetch Control Register (IbsFetchCtl)] MSRC001\_1030; and instruction execution performance controlled through [The IBS Execution Control Register (IbsOpCtl)] MSRC001\_1033. Instruction fetch sampling provides information about instruction TLB and instruction cache behavior for fetched instructions. Instruction execution sampling provides information about micro-op execution behavior. The data collected for instruction fetch performance is different from the data collected for instruction execution performance.

Instruction fetch performance is profiled by recording the following performance information (see MSRC001\_1030, MSRC001\_1031, MSRC001\_1032 for details of the events) for the tagged instruction fetch:

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

Instruction execution performance is profiled by tagging one micro-op 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 (see MSRC001\_1033, MSRC001\_1034, MSRC001\_1035, MSRC001\_1036, MSRC001\_1037, MSRC001\_1038, and MSRC001\_1039 for details of the events) is returned for the tagged micro-op:

- Branch status for branch micro-ops.
- The number clocks from when the micro-op was tagged until the micro-op retires.
- The number clocks from when the micro-op completes execution until the micro-op retires.
- Source information for DRAM, MMIO and IO access.
- L3 cache state for accesses that hit the L3 cache.
- If the operation was a load or store that missed the data cache.
- If the operation was a load or store that hit or missed the L1 and L2 TLBs.
- The linear and physical address associated with a load or store operation.



### 3 Registers

This section provides detailed field definitions for the register sets in the processor.

#### 3.1 Register Descriptions and Mnemonics

Each register in this document is referenced with a mnemonic. Each mnemonic is a concatenation of the register-space indicator and the offset of the register. Here are the mnemonics for the various register spaces:

- **IOXXX**: x86-defined input and output address space registers; XXX specifies the byte address of the IO register. This space includes IO-space configuration access registers [[The IO-Space Configuration Address Register](#)] IOCF8 and [[The IO-Space Configuration Data Port](#)] IOCFC. Accesses to these registers from each core of a node target the same registers of that node; it is not possible for a node to access these registers on a different node.
- **FYxXXX**: PCI-defined configuration space; XXX specifies the byte address of the configuration register (this may be 2 or 3 digits); Y specifies the function number; e.g., F3x40 specifies the register at function 3, address 40h. See [2.11 \[Configuration Space\]](#), for details about configuration space. There is one set of these registers per node; these registers in any node are accessible through any core of any node.
- **APICXX**: APIC memory-mapped registers; XX is the byte address offset from the base address. The base address for this space is specified by [[The APIC Base Address Register \(APIC\\_BAR\)](#)] MSR0000\_001B.
- **CPUID FnXXXX\_XXXX**: processor capabilities information returned by the CPUID instruction. See [3.9 \[CPUID Instruction Registers\]](#). Each core may only access this information for itself.
- **MSRXXXX\_XXXX**: model specific registers; XXXX\_XXXX is the MSR number. This space is accessed through x86-defined RDMSR and WRMSR instructions. There is one set of these registers per core; each core may only access its own set of these registers.

Each node includes a single set of IO-space and configuration-space registers. However, APIC, CPUID, and MSR register spaces are implemented once per processor core. Note: 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 82: Terminology in register descriptions**

Terminology	Description
Read	Capable of being read by software.
Read-only	Capable of being read but not written by software.
Write	Capable of being written by software.
Write-only	Write-only. Capable of being written by software. Reads are undefined.
Read-write	Capable of being written by software and read by software.
Set-by-hardware	Register bit is set high by hardware.
Cleared-by-hardware	Register bit is cleared low by hardware.
Updated-by-hardware	Register bit or field is updated by hardware.
Write-once	After RESET_L is asserted, these registers may be written to once. After being written, they become read-only until the next RESET_L assertion. The write-once control is byte based. So, for example, software may write each byte of a write-once DWORD as four individual transactions. As each byte is written, that byte becomes read-only.

**Table 82: Terminology in register descriptions**

Terminology	Description
Write-1-to-clear	Software must write a 1 to the bit in order to clear it. Writing a 0 to these bits has no effect.
Write-0-to-clear	Software must write a 0 to the bit in order to clear it. Writing a 1 to these bits has no effect.
Write-1-only	Software can set the bit high by writing a 1 to it. Writes of 0 have no effect. Cleared by hardware.
Reserved	Field is reserved for future use. Software is required to preserve the state read from these bits when writing to the register. Software may not depend on the state of reserved fields nor on the ability of such fields to return the state previously written.
MBZ	Must be zero. If software attempts to set an MBZ bit to 1, a general-protection exception (#GP) occurs.
RAZ	Read as zero. Writes are ignored.
GP-read	GP exception occurs on read.
GP-write	GP exception occurs on write.
GP-read-write	GP exception occurs on a read or a write.
Reset	The reset value of each register is provided below the mnemonic or in the field description. Unless otherwise noted, the register state matches the reset value when RESET_L is asserted (either a cold or a warm reset). Reset values may include: ? : a question mark in the reset value indicates that the reader should look at the bit description for reset-value details. X : an X in the reset value indicates that the field resets (warm or cold) to an unspecified state.
Cold reset	The field state is not affected by a warm reset (even if the field is labeled “cold reset: X”); it is placed into the reset state when PWROK is deasserted. See "Reset" above for the definition of characters that may be found in the cold reset value.

### 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. [MSR0000\\_0410](#)). If control of Northbridge functions is shared between software on all cores, software must ensure that only one core at a time is allowed to access the shared MSR.

## 3.2 IO Space Registers

See [3.1 \[Register Descriptions and Mnemonics\]](#) for a description of the register naming convention.

### IOCF8 IO-Space Configuration Address Register

Reset: 0000 0000h. [\[The IO-Space Configuration Address Register\] IOCF8](#), and [\[The IO-Space Configuration Data Port\] IOCFC](#), are used to access system configuration space, as defined by the PCI specification. [IOCF8](#) provides the address register and [IOCFC](#) provides the data port. Software sets up the configuration address by writing to [IOCF8](#). Then, when an access is made to [IOCFC](#), the processor generates the corresponding configuration access to the address specified in [IOCF8](#). See [2.11 \[Configuration Space\]](#).

[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 [\[The IO-Space Base/Limit](#)

Registers] F1x[DC:C0]. 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 IOCFC are translated into configuration cycles at the configuration address specified by this register. 0=IO read and write accesses are passed to the appropriate IO link and no configuration access is generated.
30:28	Reserved.
27:24	<b>ExtRegNo: extended register number.</b> Read-write. ExtRegNo provides bits[11:8] and RegNo provides bits[7:2] of the byte address of the configuration register. ExtRegNo is reserved unless it is enabled by MSRC001_001F[EnableCf8ExtCfg].
23:16	<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 IOCFC8[ExtRegNo].
1:0	Reserved.

### IOCFC IO-Space Configuration Data Port

Reset: 0000 0000h.

Bits	Description
31:0	See IOCFC8 for details about this port.

### 3.3 Function 0 HyperTransport™ Technology Configuration Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention. See 2.11 [Configuration Space] for details about how to access this space.

#### F0x00 Device/Vendor ID Register

Reset: 1200 1022h.

Bits	Description
31:16	<b>DeviceID: device ID.</b> Read-only.
15:0	<b>VendorID: vendor ID.</b> Read-only.

#### F0x04 Status/Command Register

Reset: 0010 0000h.

Bits	Description
31:16	<b>Status.</b> Read-only. Bit[20] is set to indicate the existence of a PCI-defined capability block.
15:0	<b>Command.</b> Read-only.

**F0x08 Class Code/Revision ID Register**

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.

**F0x0C Header Type Register**

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.

**F0x34 Capabilities Pointer Register**

Reset: 0000 00??h.

Bits	Description
31:8	Reserved.
7:0	<b>CapPtr: capabilities pointer.</b> Read-only. Specifies the offset of the link capabilities block based on the links that are supported by the node. The value provided is: 80h      If link 0 is supported. A0h      If link 0 is not supported and link 1 is supported. C0h      If link 0 and 1 are not supported and link 2 is supported. E0h      If link 0, 1, and 2 are not supported and link 3 is supported.

**F0x[5C:40] Routing Table Registers**

Reset: 0004 0201h. Each of these eight registers, F0x[5C, 58, 54, 50, 4C, 48, 44, 40], corresponds to a node ID for up to 8 nodes in the coherent fabric. F0x40 corresponds to node 0; F0x44 corresponds to node 1; etc. As each packet is processed by the node, it is routed to the appropriate links, or remains in the node that is processing the packet, based on the source/destination node and the type of packet being processed. The destination of requests and responses determines which of these eight registers is used to route the packet; the source of probes and broadcasts determines which of these eight registers is used to route the packet. Once the routing table register is identified, the packet is routed to the destinations based on the state of the field (in that routing table register) that corresponds to the packet type.

For each of the 9-bit fields in this register:

bit[0] = route to this node.

bit[1] = route to link 0, sublink 0.

bit[2] = route to link 1, sublink 0.

bit[3] = route to link 2, sublink 0.

bit[4] = route to link 3, sublink 0.

bit[5] = route to link 0, sublink 1.

bit[6] = route to link 1, sublink 1.

bit[7] = route to link 2, sublink 1.

bit[8] = route to link 3, sublink 1.

Bits	Description
31:27	Reserved.
26:18	<b>BCRoute: broadcast route.</b> Read-write. Specifies the routing information for broadcasts and probes.
17:9	<b>RPRoute: response route.</b> Read-write. Specifies the routing information for responses.
8:0	<b>RQRoute: request route.</b> Read-write. Specifies the routing information for requests.

### F0x60 Node ID Register

Reset: 0000 000?h.

Bits	Description
31:24	Reserved.
23:21	Must be zero. Read-write.
20:16	<b>CpuCnt[4:0]: CPU count bits[4:0].</b> Read-write. This field along with <a href="#">F0x160[CpuCnt[7:5]]</a> specifies the number of cores to be enabled in the system (the boot core of all nodes plus those cores enabled through <a href="#">F0x68[Cpu1En]</a> and <a href="#">F0x168[Cpu5En, Cpu4En, Cpu3En, Cpu2En]</a> ). Revision C and earlier: 00h = 1 core...1Fh = 32 cores; 20h through FFh are reserved. Revision D and later: 00h=1 core...2F = 48 cores; 30h through FFh are reserved. This field matches <a href="#">F0x60[NodeCnt]</a> if each nodes in the system has one core; otherwise, it would be greater than <a href="#">F0x60[NodeCnt]</a> .
15	Reserved.
14:12	<b>LkNode[2:0]: lock node ID bits[2:0].</b> Read-write. This field specifies the node ID of the node that contains the lock controller.
11	Reserved.
10:8	<b>SbNode[2:0]: Southbridge (IO hub) node ID bits[2:0].</b> Read-write. Specifies the node ID of the node that owns the link that connects to the system IO hub.
7	Reserved.
6:4	<b>NodeCnt[2:0]: node count bits[2:0].</b> Read-write. This specifies the number of coherent nodes in the system. Hardware only allows values to be programmed into this field that are consistent with the multiprocessor capabilities of the device, as specified in <a href="#">[The Northbridge Capabilities Register] F3xE8[MpCap]</a> . Attempts to write values inconsistent with the capabilities of the processor result in this field not being updated. 0h = 1 node; 1h = 2 nodes; 2h = 3 nodes; ... 7h = 8 nodes.
3	Reserved.
2:0	<b>NodeId[2:0]: node ID bits[2:0].</b> Read-write. This specifies the node ID of the node. It is reset to 0h for the boot strap processor (BSP); it is reset to 7h for all other nodes. It is expected that system configuration software programs the Node ID. The node IDs must be contiguous. For example, the node IDs in a 4-node system may be {0, 1, 2, 3}; an example of an incorrect node ID assignment in this system is {0, 1, 3, 4}. See <a href="#">MSRC001_001F[InitApicIdCpuIdLo]</a> .

### F0x64 Unit ID Register

Reset: 0000 00E0h.

Bits	Description
31:11	Reserved.

10:8	<b>SbLink: Southbridge (IO hub) link ID.</b> Read-write; set-by-hardware. This field specifies the link to which the system IO hub is connected. It is only used by the node which owns the IO hub, as indicated in <a href="#">F0x60[NodeId]</a> . For bits[9:8]: 00b = link 0; 01b = link 1; 10b = link 2; 11b = link 3. If the link is unganged, then bit[10] specifies the sublink: 0b = sublink 0; 1b = sublink 1. If the link is ganged, bit[10] is required to be low.
7:6	<b>HbUnit: host bridge Unit ID.</b> Read-only. This field 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. This field specifies the coherent link Unit ID of the memory controller.
3:2	Reserved.
1:0	<b>CpuUnit: CPU Unit ID.</b> Read-only. This field specifies the coherent link Unit ID used for core transactions.

### F0x68 Link Transaction Control Register

Reset: 0000 0000h.

Bits	Description
31	<b>EnPReqHiPriTblWlk: isoc table walk enable for posted requests.</b> Read-write. 1=Enables the use of the Isoc channel for DEV/GART table walk requests issued for base channel posted requests. To use the Isoc channel for DEV/GART requests ICFM must be enabled, one <a href="#">F3x1[54, 50, 4C, 48][IsocReqTok]</a> must be allocated on each link that can receive DEV/GART table walk requests, and one <a href="#">F3x1[54, 50, 4C, 48][IsocRspTok]</a> must be allocated on each link that can receive DEV/GART table walk responses.
30:26	Reserved.
25	<b>CHtExtAddrEn: coherent link extended address enable.</b> Read-write; however this bit is read-only, 0, for uniprocessor systems as indicated by <a href="#">F3xE8[MpCap]</a> . 1=The coherent fabric supports physical addresses of greater than 40 bits. When this bit is clear, requests to addresses above 1 terabyte result in a master abort.
24	<b>DispRefModeEn.</b> Read-write. 1=Enables support for display-refresh ordering rules. BIOS must not set this bit until display-refresh buffers have been allocated and a warm reset has occurred. See <a href="#">2.6.4.2.5 [Display Refresh And IFCM]</a> .
23	<b>InstallStateS.</b> Read-write. 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. 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. 00b = no limit. 01b = limited to 1. 10b = limited to 4. 11b = limited to 8. BIOS should set this to 10b for all products.

20	<b>SeqIdSrcNodeEn: sequence ID source node enable.</b> Read-write. 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 guaranteed 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. 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. This 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. This enables the extended APIC broadcast functionality. 0=APIC broadcast is 0Fh. 1=APIC broadcast is FFh. If this bit is set, ApicExtId must also be set.
16	<b>LintEn: local interrupt conversion enable.</b> Read-write. 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 APIC350 and APIC360. 0=ExtInt/NMI interrupts delivered unchanged.
15	<b>LimitCldtCfg: limit coherent link configuration space range.</b> Read-write. 1=Configuration accesses that (1) normally map to the configuration space within another node in the coherent fabric and (2) target a non-existent node as specified by F0x60[NodeCnt] are sent to an IO link instead. This bit should be set by BIOS once coherent fabric initialization is complete. Failure to do so may result in PCI configuration accesses to nonexistent nodes being sent into the coherent fabric, causing the system to hang.
14:13	<b>BufRelPri: buffer release priority select.</b> Read-write. Specifies the number of link DWs sent while a buffer release is pending before the buffer release is inserted into the command/data stream of a busy link. 00b = 64; 01b = 16; 10b = 8; 11b = 2. For revision C and earlier, BIOS should set this to 01b in single-link processors and 10b in multi-link processors in order to maximize link bandwidth. For revision D and later, BIOS should set this to 01b in order to maximize link bandwidth.
12	Reserved. Read-write.
11	<b>RespPassPW: response PassPW.</b> Read-write. 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. Controls probes for core-generated fills (must be 0 for multi-core or L3-cache systems; recommended to be 1 for uniprocessor, single core, no L3-cache systems). 0=Probes issued for cache fills. 1=Probes not issued for cache fills.
9	<b>DisRmtPMemC: disable remote probe memory cancel.</b> Read-write. 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. 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. 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>CPURspPassPW: CPU request PassPW.</b> Read-write. 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	<b>Cpu1En: core 1 enable.</b> Read-write. This bit and <a href="#">F0x168[Cpu5En, Cpu4En, Cpu3En and Cpu2En]</a> are used to enable each of the cores after a reset. 1=Enable the core to start fetching and executing code from the boot vector. Note: the core numbers referred to in these bits are affected by downcoring; see CpuCoreNum in section <a href="#">[The CPU Cores and Downcoring] 2.9.2</a> .
4	<b>DisMTS: disable memory controller target start.</b> Read-write. 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. 1=Disables generation of probes for core-generated, WrSized doubleword commands (must be 0 for multi-core or L3-cache systems; recommended to be 1 for uniprocessor, single core, no L3-cache systems).
2	<b>DisWrBP: disable write byte probes.</b> Read-write. 1=Disables generation of probes for core-generated, WrSized byte commands (must be 0 for multi-core or L3-cache systems; recommended to be 1 for uniprocessor, single core, no L3-cache systems).
1	<b>DisRdDwP: disable read doubleword probe.</b> Read-write. 1=Disables generation of probes for core-generated, RdSized doubleword commands (must be 0 for multi-core or L3-cache systems; recommended to be 1 for uniprocessor, single core, no L3-cache systems).
0	<b>DisRdBp: disable read byte probe.</b> Read-write. 1=Disables generation of probes for core-generated, RdSized byte commands (must be 0 for multi-core or L3-cache systems; recommended to be 1 for uniprocessor, single core, no L3-cache systems).

### F0x6C Link Initialization Control Register

Reset: 000? ???h; see individual bit definitions for reset details.

Bits	Description
20	<b>Tr0Double: Training 0 Time Double.</b> Revision C and earlier revisions: Reserved. Revision D and later: Read-write. Cold reset: 0. 1=All Training 0 times are doubled from their programmed or default/reset values. This bit should be set when using link BIST on this device connected to an ILM device, and left clear at all other times.
19:16	Must be zero. Read-write.
15:12	Reserved.
11	<b>DefLnk[2]: default link.</b> Read-only. See DefLnk[1:0], below.
10:9	<b>BiosRstDet[2:1]: BIOS reset detect bits[2:1].</b> Read-write. Cold reset: 0. See bit[5] of this register.
8	<b>DefSubLnk: default sublink.</b> Read-only. Used in conjunction with <a href="#">F0x6C[DefLnk]</a> . 0=Sublink 0. 1=Sublink 1.
7	Reserved.

6	<b>InitDet: CPU initialization command detect.</b> Read-write. 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. This bit, along with BiosRst-Det[2:1], 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	<b>DefLnk[1:0]: default link.</b> Read-only. These bits, along with DefLnk[2], above, are updated every time an incoming request is received with the link ID of the link on which the packet arrived. It is used by hardware to route packets during initialization, while F0x6C[RouteTblDisRouting]=1, and only one outstanding request is active in the system at a time. During this interval, responses are routed to the link indicated by this field. Thus, responses are properly returned to the link, or to this node, based on the source of the request. F0x6C[DefSubLnk] is used to delineate sublinks as well. <table border="0" style="margin-left: 20px;"> <tr> <td>DefLnk[2, 1:0]</td> <td>Definition</td> </tr> <tr> <td>000b</td> <td>Request came from link 0 (power-up default).</td> </tr> <tr> <td>001b</td> <td>Request came from link 1.</td> </tr> <tr> <td>010b</td> <td>Request came from link 2.</td> </tr> <tr> <td>011b</td> <td>Request came from link 3.</td> </tr> <tr> <td>100b</td> <td>Request came from a core on same node.</td> </tr> </table>	DefLnk[2, 1:0]	Definition	000b	Request came from link 0 (power-up default).	001b	Request came from link 1.	010b	Request came from link 2.	011b	Request came from link 3.	100b	Request came from a core on same node.
DefLnk[2, 1:0]	Definition												
000b	Request came from link 0 (power-up default).												
001b	Request came from link 1.												
010b	Request came from link 2.												
011b	Request came from link 3.												
100b	Request came from a core on same node.												
1	<b>ReqDis: request disable.</b> Read-write; set-by-hardware. This bit specifies if the node is allowed to generate request packets. It resets to 0 for the BSP and to 1 for all other nodes. This bit should be cleared by BIOS once the system has been initialized from the BSP. This bit is set by hardware and cleared by software. 0=Request packets may be generated. 1=Request packets may not be generated. See 2.3 [Processor Initialization].												
0	<b>RouteTblDis: routing table disable.</b> Read-write. 1=Responses are routed based on F0x6C[DefLnk] and configuration-space requests received by this node are treated as if they target this node regardless of the bus number and device number. 0=Packets are routed according to [The Routing Table Registers] F0x5C:40]. This bit is reset to 1. Once the routing tables have been set up this bit should be cleared.												

### F0x[E0, C0, A0, 80] Link Capabilities Registers

F[4, 0]x[98:80] are associated with link 0. F[4, 0]x[B8:A0] are associated with link 1. F[4, 0]x[D8:C0] are associated with link 2. F[4, 0]x[F8:E0] are associated with link 3. The function 0 registers are associated with the whole link if it is ganged or sublink 0 if it is unganged; the function 4 register are associated with sublink 1 if the link is unganged. If the node does not support a link, then the corresponding register addresses become reserved. This register is derived from the link capabilities register defined in the *HyperTransport™ I/O Link Specification*.

Bits	Description
31:29	<b>CapType: capability type.</b> Read-only, 001b.
28	<b>DropOnUnInit: drop on uninitialized link.</b> Read-only, 0.
27	<b>InbndEocErr: inbound end-of-chain error.</b> Read-only, 0.
26	<b>ActAsSlave: act as slave.</b> Read-only, 0.
25	Reserved.

24	<b>HostHide.</b> Read-only, 1.
23	<b>ChainSide.</b> Read-only, 0.
22:18	<b>DevNum: device number.</b> Read-only, 00h.
17	<b>DblEnded: double ended.</b> Read-only, 0.
16	<b>WarmReset.</b> Read-only, 1.
15:8	<b>CapPtr: capabilities pointer.</b> Read-only. Specifies the offset of the next link capabilities block based on the links that are supported by the node. Depending on which links are supported, this may be A0h, C0h, E0h, or 00h (in the case of the last link).
7:0	<b>CapID: capabilities ID.</b> Read-only. Reset: 08h. Indicates HyperTransport™ technology capability.

### F0x[E4, C4, A4, 84] Link Control Registers

F[4, 0]x[98:80] are associated with link 0. F[4, 0]x[B8:A0] are associated with link 1. F[4, 0]x[D8:C0] are associated with link 2. F[4, 0]x[F8:E0] are associated with link 3. The function 0 registers are associated with the whole link if it is ganged or sublink 0 if it is unganged; the function 4 register are associated with sublink 1 if the link is unganged. If the node does not support a link, then the corresponding register addresses become reserved. This register is derived from the link control register defined in the *HyperTransport™ I/O Link Specification*

Bits	Description												
31	Reserved.												
30:28	<p><b>WidthOut: link width out.</b> Read-write. Cold reset: (see text below). Specifies the operating width of the outgoing link. Legal values are:</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Link width</th> <th>Bits</th> <th>Link width</th> </tr> </thead> <tbody> <tr> <td>001b</td> <td>16 bits</td> <td>111b</td> <td>not connected</td> </tr> <tr> <td>000b</td> <td>8 bits</td> <td></td> <td></td> </tr> </tbody> </table> <p>The cold reset value of this field depends on the widths of the links of the connecting device, per the link specification. This field cannot be set to 16 bits when reganging a link until F0x[18C:170][Ganged] has been set to 1. This field cannot be changed by software if the link was determined to be disconnected by hardware at cold reset. Note: After this field is written by software, the link width does not change until either a warm reset or a link disconnect sequence occurs through LDTSTOP_L.</p>	Bits	Link width	Bits	Link width	001b	16 bits	111b	not connected	000b	8 bits		
Bits	Link width	Bits	Link width										
001b	16 bits	111b	not connected										
000b	8 bits												
27	Reserved.												
26:24	<p><b>WidthIn: link width in.</b> Read-write. Cold reset: (see text below). Specifies the operating width of the incoming link. See F0x[E4, C4, A4, 84][WidthOut] for legal values. The cold reset value of this field depends on the widths of the links of the connecting device, per the link specification. This field cannot be set to 16 bits when reganging a link until F0x[18C:170][Ganged] has been set to 1. This field cannot be changed by software if the link was determined to be disconnected by hardware at cold reset. Note: After this field is written to by software, the link width does not change until either a warm reset or a link disconnect sequence occurs through LDTSTOP_L.</p>												
23	Reserved.												
22:20	<p><b>MaxWidthOut: max link width out.</b> Read-only. This specifies the width of the outgoing link to be 8 bits or 16 bits wide, depending on the processor version. See F0x[E4, C4, A4, 84][WidthOut] for the encoding. Note: this indicates an 8-bit link if the link is unganged.</p>												
19	Reserved.												

18:16	<b>MaxWidthIn: max link width in.</b> Read-only. This specifies the width of the incoming link to be 8 bits or 16 bits wide, depending on the processor version. See <a href="#">F0x[E4, C4, A4, 84][WidthOut]</a> for the encoding. Note: this indicates an 8-bit link if the link is ungangged.
15	<b>Addr64BitEn: 64-bit address packet enable.</b> Read-write. Cold reset: 0. 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. Note: <a href="#">F0x68[CHtExtAddrEn]</a> is required to be set if this bit is set for any IO link. Note: the link specification indicates that this bit is cleared by a <i>warm</i> 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	<b>ExtCTL: extended control time during initialization.</b> Read-write. Cold reset: 0. This specifies the time in which the link CTL signal is held asserted during the initialization sequence that follows an LDTSTOP_L deassertion, after CTL is detected asserted. 0=At least 16 bit times. 1=About 50 microseconds. This bit is ignored at Gen3 frequencies.
13	<b>LdtStopTriEn: LDTSTOP tristate enable.</b> Read-write. Cold reset: 0. This bit is ignored by hardware when the link is operating at Gen3 frequencies. 1=During the LDTSTOP_L disconnect sequence, the link transmitter signals are placed into the high-impedance state and the receivers are prepared for the high-impedance mode. For the receivers, this includes cutting power to the receiver differential amplifiers and ensuring that there are no resultant high-current paths in the circuits. 0=During the LDTSTOP_L disconnect sequence, the link transmitter signals are driven, but in an undefined state, and the link receiver signals are assumed to be undriven. BIOS should set this bit to 1 for revision C and later revisions if the connecting device supports LdtStopTriEn=1.
12	<b>IsocEn: isochronous flow-control mode enable.</b> Read-write. Cold reset: 0. This bit is set to place the link into isochronous flow-control mode (IFCM), as defined by the link specification. However, the flow-control mode does not change until a warm reset occurs. 1=IFCM. 0=Normal flow-control mode. Note: all coherent links of the system must use the same flow-control mode. See <a href="#">2.6.4.2.5 [Display Refresh And IFCM]</a> .
11:10	Reserved.
9:8	<b>CrcErr: CRC Error.</b> Read; set-by-hardware; write-1-to-clear. Cold reset: 00b. Bit[1] applies to the upper byte of the link and bit[0] applies to the lower byte. 1=The hardware detected a CRC error on the incoming link while not in retry mode; if in retry mode, then bit[8] may be set to indicate an uncorrectable error was detected; such uncorrectable error cases are: <ul style="list-style-type: none"> <li>• Link reconnect fails exceeding the limit in <a href="#">[The Link Global Retry Control Register]</a> <a href="#">F0x150[TotalRetryAttempts]</a>.</li> </ul>

7	<p><b>TransOff: transmitter off.</b> Revision D0 and earlier: Reserved. Revision D1 and later: Read-write. 1=Output tristates. Notes about TransOff and EndOfChain:</p> <ul style="list-style-type: none"> <li>• The initial state is programmed by hardware after each cold reset (0 if the link connects; 1 if it does not connect; see 2.7.1.3 [Link Type Detect]).</li> <li>• If F0x16C[ConnDly]=0: setting these bits causes the link behavior to change immediately; clearing these bits is illegal; warm resets and LDTSTOPS do not affect the state of these bits; TransOff may only be set if EndOfChain is set.</li> <li>• If F0x16C[ConnDly]=1: values written to these bits take effect during the next warm reset or LDTSTOP. Reads represent the last value written rather than the current link behavior; TransOff and EndOfChain should always be programmed to the same state. For dual-node processors, BIOS should program this according to section 2.7.1.5 [Link Mapping for Dual-Node Processors in the G34 Package].</li> </ul>
6	<p><b>EndOfChain: end of chain.</b> Read-write. 1=The link is not part of the logical HyperTransport™ technology chain; packets targeting this link are dropped. See TransOff, above. For dual-node processors, BIOS should program this according to section 2.7.1.5 [Link Mapping for Dual-Node Processors in the G34 Package].</p>
5	<p><b>InitComplete: initialization complete.</b> Read-only; set-by-hardware. Reset: 0. This bit is set by hardware when low-level link initialization has successfully completed. If there is no device on the other end of the link, or if the device on the other side of the link is unable to properly perform link initialization, then the bit is not set. This bit is not cleared for LDTSTOP# disconnects or retries. Hardware may report 0 during BIST mode or ILM.</p>
4	<p><b>LinkFail: link failure.</b> Read; set-by-hardware; write-1-to-clear. Cold reset: 0. This bit is set high by the hardware when a CRC error is detected on the link (if enabled by CrcFloodEn), the link fails to reconnect, if a sync flood is received by the link, or if the link is not used in the system. See 2.12.1.5.1 [Common Diagnosis Information].</p>
3	<p><b>CrcForceErr: CRC force error command.</b> Read-write. Reset: 0. 1=The link transmission logic generates erroneous periodic or per-packet CRC values on all enabled byte lanes. 0=Transmitted CRC values match the values calculated per the link specification. This bit is intended to be used to check the CRC failure detection logic of the device on the other side of the link. See F0x150[ForceErrType] for retry mode.</p>
2	Reserved.
1	<p><b>CrcFloodEn: CRC flood enable.</b> Read-write. Reset: 0. 1=Setting either of the CrcErr bits results in sync packets to all enabled outgoing links and the F0x[E4, C4, A4, 84][LinkFail] bit is set. 0=Setting either of the CrcErr bits do not result in sync packets or setting the F0x[E4, C4, A4, 84][LinkFail] bit. In Gen3 protocol, exceeding the F0x150[TotalRetryAttempts] limit results in a sync flood regardless of how CrcFloodEn is set. The resulting sync flood does not propagate to other links or set Linkfail unless CrcFloodEn is set. This bit is ignored if F3x44[SyncPktGenDis] is set.</p>
0	Reserved.

### F0x[E8, C8, A8, 88] Link Frequency/Revision Registers

F[4, 0]x[98:80] are associated with link 0. F[4, 0]x[B8:A0] are associated with link 1. F[4, 0]x[D8:C0] are associated with link 2. F[4, 0]x[F8:E0] are associated with link 3. The function 0 registers are associated with the whole link if it is ganged or sublink 0 if it is unganged; the function 4 register are associated with sublink 1 if the link is unganged. If the node does not support a link, then the corresponding register addresses become reserved. This register is derived from the link frequency/revision register defined in the *HyperTransport™ I/O*

*Link Specification*

Bits	Description																								
31:16	<p><b>LnkFreqCap: link frequency capability.</b> Read-only. Reset: values vary with product. These bits, along with <b>F0x[FC, DC, BC, 9C][FreqCapExt]</b> for revision D and later, indicate which link frequencies the processor supports. The bits are encoded as: 1=The link frequency is supported; 0=The link frequency is not supported. The bits correspond to different link frequencies as follows:</p> <table> <tr> <td>Bit 0: 200 MHz (this bit is 1 in all products).</td> <td>Bit 8: 1400 MHz.</td> </tr> <tr> <td>Bit 1: 300 MHz (this bit is 0 in all products).</td> <td>Bit 9: 1600 MHz.</td> </tr> <tr> <td>Bit 2: 400 MHz.</td> <td>Bit 10: 1800 MHz.</td> </tr> <tr> <td>Bit 3: 500 MHz (this bit is 0 in all products).</td> <td>Bit 11: 2000 MHz.</td> </tr> <tr> <td>Bit 4: 600 MHz.</td> <td>Bit 12: 2200 MHz.</td> </tr> <tr> <td>Bit 5: 800 MHz.</td> <td>Bit 13: 2400 MHz.</td> </tr> <tr> <td>Bit 6: 1000 MHz.</td> <td>Bit 14: 2600 MHz.</td> </tr> <tr> <td>Bit 7: 1200 MHz.</td> <td>Bit 15: reserved.</td> </tr> </table> <p>This field indicates logical support for these frequencies; however, electrical support for these frequencies may vary based on the part number and other system considerations.</p>	Bit 0: 200 MHz (this bit is 1 in all products).	Bit 8: 1400 MHz.	Bit 1: 300 MHz (this bit is 0 in all products).	Bit 9: 1600 MHz.	Bit 2: 400 MHz.	Bit 10: 1800 MHz.	Bit 3: 500 MHz (this bit is 0 in all products).	Bit 11: 2000 MHz.	Bit 4: 600 MHz.	Bit 12: 2200 MHz.	Bit 5: 800 MHz.	Bit 13: 2400 MHz.	Bit 6: 1000 MHz.	Bit 14: 2600 MHz.	Bit 7: 1200 MHz.	Bit 15: reserved.								
Bit 0: 200 MHz (this bit is 1 in all products).	Bit 8: 1400 MHz.																								
Bit 1: 300 MHz (this bit is 0 in all products).	Bit 9: 1600 MHz.																								
Bit 2: 400 MHz.	Bit 10: 1800 MHz.																								
Bit 3: 500 MHz (this bit is 0 in all products).	Bit 11: 2000 MHz.																								
Bit 4: 600 MHz.	Bit 12: 2200 MHz.																								
Bit 5: 800 MHz.	Bit 13: 2400 MHz.																								
Bit 6: 1000 MHz.	Bit 14: 2600 MHz.																								
Bit 7: 1200 MHz.	Bit 15: reserved.																								
15:12	Reserved.																								
11:8	<p><b>Freq[3:0]: link frequency.</b> Read-write. Cold reset: 0h. This field and <b>F0x[FC, DC, BC, 9C][Freq[4]]</b> specifies the link frequency. Note Freq[4] is only valid for revision D and later processors. Legal values are:</p> <table> <tr> <td>0h: 200 MHz.</td> <td>8h: 1400 MHz.</td> <td>10h: reserved.</td> </tr> <tr> <td>1h: Reserved.</td> <td>9h: 1600 MHz.</td> <td>11h: 2800 MHz.</td> </tr> <tr> <td>2h: 400 MHz.</td> <td>Ah: 1800 MHz.</td> <td>12h: 3000 MHz.</td> </tr> <tr> <td>3h: Reserved.</td> <td>Bh: 2000 MHz.</td> <td>13h: 3200 MHz.</td> </tr> <tr> <td>4h: 600 MHz.</td> <td>Ch: 2200 MHz.</td> <td>14h-1Fh: Reserved.</td> </tr> <tr> <td>5h: 800 MHz.</td> <td>Dh: 2400 MHz.</td> <td></td> </tr> <tr> <td>6h: 1000 MHz.<sup>1</sup></td> <td>Eh: 2600 MHz.</td> <td></td> </tr> <tr> <td>7h: 1200 MHz.</td> <td>Fh: Reserved.</td> <td></td> </tr> </table> <p>After this field is updated, the link frequency does not change until either a warm reset or a link disconnect sequence occurs through LDTSTOP_L. The value read from this field is the last value written. Writes to this field are ignored if a non-supported frequency is written.</p> <p>Note: 1. Must not be used for links connecting internal nodes as specified by <b>F0x1A0[IntLnkRoute]</b>.</p>	0h: 200 MHz.	8h: 1400 MHz.	10h: reserved.	1h: Reserved.	9h: 1600 MHz.	11h: 2800 MHz.	2h: 400 MHz.	Ah: 1800 MHz.	12h: 3000 MHz.	3h: Reserved.	Bh: 2000 MHz.	13h: 3200 MHz.	4h: 600 MHz.	Ch: 2200 MHz.	14h-1Fh: Reserved.	5h: 800 MHz.	Dh: 2400 MHz.		6h: 1000 MHz. <sup>1</sup>	Eh: 2600 MHz.		7h: 1200 MHz.	Fh: Reserved.	
0h: 200 MHz.	8h: 1400 MHz.	10h: reserved.																							
1h: Reserved.	9h: 1600 MHz.	11h: 2800 MHz.																							
2h: 400 MHz.	Ah: 1800 MHz.	12h: 3000 MHz.																							
3h: Reserved.	Bh: 2000 MHz.	13h: 3200 MHz.																							
4h: 600 MHz.	Ch: 2200 MHz.	14h-1Fh: Reserved.																							
5h: 800 MHz.	Dh: 2400 MHz.																								
6h: 1000 MHz. <sup>1</sup>	Eh: 2600 MHz.																								
7h: 1200 MHz.	Fh: Reserved.																								
7:0	<p><b>Revision.</b> Read-only, 60h. Indicates that the processor is designed to version 3.00 of the link specification.</p>																								

**F0x[EC, CC, AC, 8C] Link Feature Capability Registers**

F[4, 0]x[98:80] are associated with link 0. F[4, 0]x[B8:A0] are associated with link 1. F[4, 0]x[D8:C0] are associated with link 2. F[4, 0]x[F8:E0] are associated with link 3. The function 0 registers are associated with the whole link if it is ganged or sublink 0 if it is unganged; the function 4 register are associated with sublink 1 if the link is unganged. If the node does not support a link, then the corresponding register addresses become reserved. This register is derived from the link feature capability register defined in the *HyperTransport™ I/O Link Specification*. Unless otherwise specified: 0=The feature is not supported; 1=The feature is supported.

Bits	Description
31:10	Reserved.
9	<b>UpstrCfgCap: upstream configuration capable.</b> Read-only, 0.



8	<b>ExtRegSet: extended register set.</b> Read-only, 0.
7:6	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	<b>64BitAddr: 64-bit link addressing.</b> Read-only, 1.
3	<b>ExtCTLRqd: extended CTL required.</b> Read-only, 0.
2	<b>CrcTstMode: CRC test mode.</b> Read-only, 0.
1	<b>LdtStopMode: LDTSTOP supported.</b> Read-only, 1.
0	<b>IsocMode: isochronous flow control mode.</b> Read-only, 1.

### **F0x[F0, D0, B0, 90] Link Base Channel Buffer Count Registers**

F[4, 0]x[98:80] are associated with link 0. F[4, 0]x[B8:A0] are associated with link 1. F[4, 0]x[D8:C0] are associated with link 2. F[4, 0]x[F8:E0] are associated with link 3. The function 0 registers are associated with the whole link if it is ganged or sublink 0 if it is unganged; the function 4 register are associated with sublink 1 if the link is unganged. If the node does not support a link, then the corresponding register addresses become reserved. If a link does not initialize properly or is not connected, then the reset state of the buffer count fields is X. See 2.7.1.3.1 [Unused Links].

F0x[F0, D0, B0, 90] and F0x[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, or reallocated by BIOS to the hard-allocated buffer counts. When the link initializes, the default number of buffers hard-allocated to each virtual channel in F0x[F0, D0, B0, 90] differs based on whether the link initializes to IO or coherent protocol, ganged or unganged, as follows (if LockBc=0):

	<u>FreeData</u>	<u>FreeCmd</u>	<u>RspData</u>	<u>NpReqData</u>	<u>ProbeCmd</u>	<u>RspCmd</u>	<u>PReq</u>	<u>NpReqCmd</u>
IO link ganged:	4	16	2	2	0	4	8	36
Coh link ganged:	4	16	4	4	18	18	4	8
IO link unganged:	4 <sup>1</sup>	16 <sup>1</sup>	1	1	0	2	4	18
Coh link unganged:	4 <sup>1</sup>	16 <sup>1</sup>	2	2	9	9	2	4

Note:

1. For revision D and later, the number of buffers is half the number indicated.

The cold-reset register state (ganged or unganged) is: IO link=0285\_0292h; coherent link=028A\_9944h.

For all fields on revision D and later or all fields except for FreeData and FreeCmd on revision C and earlier revisions, 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.

For revision C and earlier revisions: for the FreeData and FreeCmd fields, the number of buffers allocated is 2 times the value of the field, whether the link is ganged or unganged. If the link is unganged, then the FreeData and FreeCmd counts are provided in the corresponding register of function 0; these fields in the corresponding register of function 4 are reserved; these pools of buffers are applied to both sublinks as needed.

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.

After boot, the allocation may be changed by BIOS. Rules governing these registers are as follows:

- Base channel buffers are specified in `F0x[F0, D0, B0, 90]`; isochronous buffer counts (if in IFCM) are specified in `F0x[F4, D4, B4, 94]`.
- New values written to these registers take effect after a warm reset if LockBc is set. If LockBc is clear, these registers return to their default values after a warm reset.
- The total number of command buffers allocated in the base and isochronous registers of a link cannot exceed 64.
- For ungangled links on revision D and later, the total number of command buffers allocated in the base and isochronous registers of each sublink cannot exceed 32.
- The total number of data buffers allocated in the base and isochronous registers of a link cannot exceed 16.
- For ungangled links on revision D and later, the total number of data buffers allocated in the base and isochronous registers of each sublink cannot exceed 8.
- The total number of hard allocated command buffers (ProbeCmd, RspCmd, PReq, NpReqCmd, and `F0x[F4, D4, B4, 94]`[IsocRspCmd, IsocPReq, and IsocNpReqCmd]) cannot exceed 48.
- For ungangled links on revision D and later, the total number of hard allocated command buffers (ProbeCmd, RspCmd, PReq, NpReqCmd, and `F0x[F4, D4, B4, 94]`[IsocRspCmd, IsocPReq, and IsocNpReqCmd]) for each sublink cannot exceed 24.
- If ungangled on revision C and earlier revisions, the total buffer counts of a link are shared between the two sublinks.
- If ungangled on revision C and earlier revisions, the free command and free data buffer pools are shared between the two sublinks of a link and are specified by `F0x[F0, D0, B0, 90]`.
- If ungangled on revision D and later, the total buffer counts of a link are divided in half with each half allocated to each of the two sublinks.
- The isochronous buffer counts (`F0x[F4, D4, B4, 94]`) default to zero. BIOS must set up non-zero counts (and adjust the base channel counts accordingly) prior to enabling IFCM.
- If an IOMMU is present in the system, `F0x[F4, D4, B4, 94]`[IsocNpReqCmd] must be non-zero for all enabled links and `F0x[F4, D4, B4, 94]`[IsocNpReqData] must be non-zero for all enabled coherent links.

See 2.6.4.2.5.1 [Recommended Buffer Count Settings for Single Link Processors] and 2.6.4.2.5.2 [Recommended Buffer Count Settings for Multi-link Processors].

Bits	Description
31	<b>LockBc: lock buffer count register.</b> Read-write. Cold reset: 0. 1=The buffer count registers, <code>F0x[F0, D0, B0, 90]</code> and <code>F0x[F4, D4, B4, 94]</code> 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.
30:28	Reserved
27:25	<b>FreeData: free data buffer count.</b> Read-write.
24:20	<b>FreeCmd: free command buffer count.</b> Read-write.
19:18	<b>RspData: response data buffer count.</b> Read-write.
17:16	<b>NpReqData: non-posted request data buffer count.</b> Read-write.
15:12	<b>ProbeCmd: probe command buffer count.</b> Read-write.

11:8	<b>RspCmd: response command buffer count.</b> Read-write.
7:5	<b>PReq: posted request command and data buffer count.</b> Read-write. This specifies the number of posted command and posted data buffers allocated.
4:0	<b>NpReqCmd: non-posted request command buffer count.</b> Read-write.

### **F0x[F4, D4, B4, 94] Link Isochronous Channel Buffer Count Registers**

Reset: 0000 0000h. F[4, 0]x[98:80] are associated with link 0. F[4, 0]x[B8:A0] are associated with link 1. F[4, 0]x[D8:C0] are associated with link 2. F[4, 0]x[F8:E0] are associated with link 3. The function 0 registers are associated with the whole link if it is ganged or sublink 0 if it is unganged; the function 4 register are associated with sublink 1 if the link is unganged. If the node does not support a link, then the corresponding register addresses become reserved. If a link does not initialize properly or is not connected, then the reset state of the buffer count fields is X. See F0x[F0, D0, B0, 90] for information about the buffer count fields. See 2.7.1.3.1 [Unused Links].

Bits	Description
31:29	Reserved.
28:27	<b>IsocRspData: isochronous response data buffer count.</b> Read-write.
26:25	<b>IsocNpReqData: isochronous non-posted request data buffer count.</b> Read-write.
24:22	<b>IsocRspCmd: isochronous response command buffer count.</b> Read-write.
21:19	<b>IsocPReq: isochronous posted request command and data buffer count.</b> Read-write. This specifies the number of isochronous posted command and posted data buffers allocated.
18:16	<b>IsocNpReqCmd: isochronous non-posted request command buffer count.</b> Read-write.
15:8	<b>SecBusNum: secondary bus number.</b> Read-write. This 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 [The Configuration Map Registers] F1x[EC:E0][BusNumBase] field of the node (unless F1x[EC:E0][DevCmpEn]=1, in which case this field should be 00h).
7:0	Reserved.

### **F0x[F8, D8, B8, 98] Link Type Registers**

Reset: 0000 00??h. F[4, 0]x[98:80] are associated with link 0. F[4, 0]x[B8:A0] are associated with link 1. F[4, 0]x[D8:C0] are associated with link 2. F[4, 0]x[F8:E0] are associated with link 3. The function 0 registers are associated with the whole link if it is ganged or sublink 0 if it is unganged; the function 4 registers are associated with sublink 1 if the link is unganged. If the node does not support a link, then the corresponding register addresses become reserved.

Bits	Description
31:5	Reserved.
4	<b>LinkConPend: link connect pending.</b> Read-only. 1=Hardware is currently determining if the link is connected to another device. 0=The link connection has been determined. This bit qualifies the Link-Con bit.
3	Reserved.
2	<b>NC: non coherent.</b> Read-only. This bit specifies the link type. 0=coherent link. 1=IO link.

1	<b>InitComplete: initialization complete.</b> Read-only. 1=Link initialization is complete. This is a duplicate of [The Link Control Registers] F0x[E4, C4, A4, 84][InitComplete]. The NC bit is invalid until link initialization is complete.
0	<b>LinkCon: link connected.</b> Read-only. 1=The link is connected to another device. 0=The link is not connected. This is not valid until LinkConPend=0.

### **F0x[FC, DC, BC, 9C] Link Frequency Extension Registers**

Revision C and earlier: Reserved. Revision D and later: F[4, 0]x[9C:80] are associated with link 0. F[4, 0]x[BC:A0] are associated with link 1. F[4, 0]x[DC:C0] are associated with link 2. F[4, 0]x[FC:E0] are associated with link 3. The function 0 registers are associated with the whole link if it is ganged or sublink 0 if it is unganged; the function 4 registers are associated with sublink 1 if the link is unganged. If the node does not support a link, then the corresponding register addresses become reserved. This register is derived from the link frequency extension register defined in the *HyperTransport™ I/O Link Specification*

Bits	Description
31:16	Reserved.
15:1	<b>FreqCapExt: link frequency capability extension.</b> Read-only. Reset: values vary with product. These bits along with F0x[E8, C8, A8, 88][LnkFreqCap] indicate which additional link frequencies the processor supports. The bits are encoded as: 1=The link frequency is supported; 0=The link frequency is not supported. The bits correspond to different link frequencies as follows: Bit 0: 2.8 GHz. Bit 1: 3.0 GHz. Bit 2: 3.2 GHz. Bits 3-14: reserved. This field indicates logical support for these frequencies; however, electrical support for these frequencies may vary based on the part number and other system considerations.
0	<b>Freq[4]: link frequency.</b> Read-write. Cold reset: 0. See F0x[E8, C8, A8, 88][Freq[3:0]] for the definition of this bit. If Freq[4] is being changed, a write to F0x[FC, DC, BC, 9C] must be followed by a write to the F0x[E8, C8, A8, 88] register corresponding to the same link (or sublink).

### **F0x[11C, 118, 114, 110] Link Clumping Enable Registers**

Reset: 0000 0000h. F0x[120, 110] is associated with link 0; F0x[124, 114] is associated with link 1; F0x[128, 118] is associated with link 2. F0x[12C, 11C] is associated with link 3. F0x[11C, 118, 114, 110] are associated with the whole link if it is ganged or sublink 0 if it is unganged; F0x[12C, 128, 124, 120] are associated with sublink 1 if the link is unganged. If the node does not support a link, then the corresponding register addresses become 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
31:2	<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.
1:0	Reserved.

### **F0x[12C, 128, 124, 120] Sublink 1 Clumping Enable Registers**

Reset: 0000 0000h. See F0x[11C, 118, 114, 110]. If a link is ganged or not supported, then the corresponding register in this group is reserved.

### **F0x[14C:130] Link Retry Registers**

The following retry registers associated with the following links are specified here:

F0x130: link 0, sublink 0	F0x140: link 0, sublink 1
F0x134: link 1, sublink 0	F0x144: link 1, sublink 1
F0x138: link 2, sublink 0	F0x148: link 2, sublink 1
F0x13C: link 3, sublink 0	F0x14C: link 3, sublink 1

If a link is ganged, then the sublink 0 retry register specifies the whole link retry register function and the sublink 1 retry register is reserved. If a link is not supported by the node, then both the sublink 0 and sublink 1 retry registers are reserved. These registers are reserved if F3xE8[LnkRtryCap]=0.

Bits	Description
31:16	<b>RetryCount.</b> Read-write. Cold reset: 0. This is a 16-bit counter that is incremented by hardware. The counter is incremented in two ways, (1) the counter increments once for each failed training attempt and (2) the counter increments once for each packet error that causes a retry attempt. If the counter value is FFFFh it increments to 0000h and the RetryCountRollover bit is set. RetryCount is not incremented for retries initiated by other devices, only for errors detected by the node.
15:13	Reserved.
12	<b>DataCorruptOut: sent corrupted data.</b> Read; write-1-to-clear. Cold reset: 0. 1=Data sent on the link was marked with Data Error to indicate that it is known to be corrupted.
11	<b>InitFail.</b> Read; write-1-to-clear. Cold reset: 0. 1=Initialization sequence failed on a link reconnect.
10	<b>StompedPktDet: stomped packet detected by receiver.</b> Read; write-1-to-clear. Cold reset: 0.
9	<b>RetryCountRollover.</b> Read; write-1-to-clear. Cold reset: 0. See RetryCount.
8	<b>RetryErrorDet: retry error detected.</b> Read; write-1-to-clear. Cold reset: 0. 1=A retry was initiated in one of the ways listed in RetryCount.
7:6	<b>ShortRetryAttempts.</b> Read-write. Reset: 11b. This specifies the number of short retry attempts when operating at a Gen3 link frequency; after exceeding this value, long retries are attempted until the max count specified by [The Link Global Retry Control Register] F0x150[TotalRetryAttempts] is exceeded. The retry attempt counter is not incremented for retries initiated by other devices, only for errors detected by the node. This field is ignored when operating at Gen1 link frequencies.
5:4	Reserved.
3	<b>DisRetryDataError: disable link retry on data packet error.</b> Read-write. Reset: 0. 1=The node does not initiate the retry sequence if an error is detected on a data packet; Data packets are acknowledged even if there is a CRC error. This is intended to support debug modes in which errors are detected but allowed to propagate through the crossbar in order to allow logging of error data patterns in trace mode.
2	<b>DisRetryAnyError: disable link retry on any packet error.</b> Read-write. Reset: 0. 1=The node does not initiate the retry sequence if an error is detected; Packets are acknowledged even if there is a CRC error. This is intended to support debug modes in which errors are detected but allowed to propagate through the crossbar in order to allow logging of error data patterns in trace mode.

1	<b>ForceRetryError.</b> Read-write; cleared-by-hardware once the error has been injected onto the link. Reset: 0. This bit may be used by diagnostic software to test the error detection and retry logic of the link. 1=Forces a CRC error in one packet from the transmitter. See [The Link Global Retry Control Register] F0x150[ForceErrType, MultRetryErr].
0	<b>RetryModeEnable.</b> Read-write; changes take effect on the next warm reset. Cold reset: 0. 1=Place the link in error retry mode when reconnecting after the next warm reset. See Table 32.

### F0x150 Link Global Retry Control Register

This register is reserved if F3xE8[LnkRtryCap]=0. All fields of this register are expected to be programmed the same in all nodes of the system (except ForceErrType and MultRetryErr). Feature support varies by revision. See Table 2.

Bits	Description
31:19	Reserved.
18:16	<b>TotalRetryAttempts.</b> Read-write. Cold reset: 7h if F3xE8[LnkRtryCap]=1; 0h if F3xE8[LnkRtryCap]=0. Specifies the total number of retry attempts (short and long) allowed on any link before the link is considered to have failed. When operating at Gen3 link frequencies, short retry attempts are limited by [The Link Retry Registers] F0x[14C:130][ShortRetryAttempts]; the remaining are long retry attempts. The link is determined to have failed after TotalRetryAttempts + 1 errors; e.g., if TotalRetryAttempts=7, then the link is determined to have failed as a result of the 8 errors. This register should be programmed to values of 1 or greater. The retry attempt counter for a link is incremented each time F0x[14C:130][RetryCount] for that link is incremented.
15:14	Reserved.
13	<b>HtRetryCrcDatInsDynEn: link retry CRC data insertion enable.</b> Read-write. Cold reset: 0. 1=Enables dynamic mode for CRC insertion in data packets on a coherent link. In this mode, the transmitter follows the insertion policy defined by HtRetryCrcDatIns[2:0] for a link which is close to idle; however, it inserts fewer CRC cells as the link becomes busy. BIOS should set this bit.
12	<b>HtRetryCrcCmdPackDynEn: link retry CRC command packet dynamic mode enable.</b> Read-write. Cold reset: 0. 1=Enables dynamic mode for CRC command packing on a coherent link with retry enabled. In this mode, command packing is suspended when a link is not busy. Setting this bit is only effective if HtRetryCrcCmdPack is also set. BIOS should set this bit.
11:9	<b>HtRetryCrcDatIns: link retry CRC data insertion.</b> Read-write. Cold reset: 0. Specifies insertion of additional CRC cells in a data packet over coherent link. For a data packet defined as a data command header followed by at most 4 data beats (beat 0 through beat 3) of 16 bytes each with a data packet CRC at the end, this bit is defined as follows: 000b no additional CRC insertion 001b CRC insertion after data beat 0 010b CRC insertion after cmd header and after data beat 0 011b CRC insertion after cmd header, data beat 0 and data beat 1 100b CRC insertion after cmd header, data beat 0, data beat 1 and data beat 2 101b - 111b reserved BIOS should set this to 100b.
8	<b>HtRetryCrcCmdPack: link retry CRC command packing.</b> Read-write. Cold reset: 0. 1=Enables command packing on coherent links with retry enabled. Command packing allows a coherent link transmitter to pack multiple commands together with a single CRC. BIOS should set this bit.
7	Reserved.

6:5	<b>ForceErrType: force error type.</b> Read-write. Cold reset: 00b. Specifies the error type generated by F0x[14C:130][ForceRetryError], F0x[E4, C4, A4, 84][CrcForceErr], and F3x44[GenCrcErrByte1, GenCrcErrByte0]. 00b Forces per-packet CRC error in any packet type (NOP, command, or data). 01b Forces per-packet CRC error on a command packet only (not including NOP). 10b Forces per-packet CRC error on a data packet only. If HtRetryCrcDatIns=1, then the error is forced into the first CRC of the packet. 11b Forces per-packet CRC error on a data packet only. If HtRetryCrcDatIns=1, then the error is forced into the last CRC of the packet.
4	<b>MultRetryErr: multiple retry force error.</b> Read-write. Cold reset: 0. 1=Inhibits hardware clearing of [The Link Retry Registers] F0x[14C:130][ForceRetryError], thereby causing multiple link retry errors (at a very high rate). This can be used to test software associated with reporting of multiple link reconnect failures.
3:0	Reserved.

### F0x160 Extended Node ID Register

Revision C and earlier: Reserved. Revision D and later: Reset: 0000 0000h.

Bits	Description
31:19	Reserved.
18:16	<b>CpuCount[7:5]: CPU count bits[7:5].</b> Read-write. See F0x60[CpuCount[4:0]].
15:0	Reserved.

### F0x164 Coherent Link Traffic Distribution Register

Reset: 0000 0000h. See 2.6.4.2.4 [Link Traffic Distribution] for details about link traffic distribution.

Bits	Description
31:24	Reserved.
23:16	<b>DstLnk[7:0]: distribution destination link.</b> Read-write. Specifies the pool of links over which traffic is distributed. Note that packets which are not eligible for distribution (for example sized reads and writes) are routed normally, based on the routing tables. If the link is ganged, then only the sublink 0 bit need be set; the sublink 1 bit is ignored. bit 0 - link 0, sublink 0                      bit 4 - link 0, sublink 1 bit 1 - link 1, sublink 0                      bit 5 - link 1, sublink 1 bit 2 - link 2, sublink 0                      bit 6 - link 2, sublink 1 bit 3 - link 3, sublink 0                      bit 7 - link 3, sublink 1
15:11	Reserved.
10:8	<b>DstNode[2:0]: coherent link distribution destination node.</b> Read-write. For cHTReqDistEn and cHTRspDistEn, DstNode specifies the destination node for which coherent link traffic should be distributed. For cHTPrbDistEn, all probes originating from the local node are distributed irrespective of the value of DstNode. Packets specified by cHTReqDistEn, cHTRspDistEn, and cHTPrbDistEn that are destined for DstNode are distributed between links specified by DstLnk in approximately a round-robin fashion.
7:3	Reserved.



2	<b>cHTPrbDistEn: coherent link probe distribution enable.</b> Read-write. Enables coherent link traffic distribution for the probe virtual channel. The probes affected by this bit are limited to broadcast probes sourced from the local node (as opposed to being forwarded from another node).
1	<b>cHTRspDistEn: coherent link response distribution enable.</b> Read-write. 1=Enables coherent link traffic distribution for the response virtual channel. The responses affected by this bit are limited to responses to the request types listed for cHTReqDistEn and which are sourced from the local node (as opposed to being forwarded from another node).
0	<b>cHTReqDistEn: coherent link request distribution enable.</b> Read-write. 1=Enable coherent link traffic distribution for the request virtual channel. The requests affected by this bit are limited to cache block transactions and directed probes which are sourced from the local node (as opposed to being forwarded from another node).

### F0x168 Extended Link Transaction Control Register

Reset: 0000 0000h.

Bits	Description
31:11	Reserved.
10	<b>DisNcHtCmdThrottle: disable IO link command throttling.</b> Read-write. 0=The node limits generation of the first DWORD of link-defined commands to no more than one every four DWORDs of link bandwidth. If, for example, a 2-DWORD command is transmitted by the node, and there is no data that follows, then the node sends at least 2 DWORDs of NOPs (possibly including buffer release credits) before generating the next command packet. This bit applies to both Gen1 and Gen3 frequencies and protocols. This bit does not affect coherent links. Some IO devices may require this bit to be clear. 1=The node does not limit the rate at which commands are generated on IO links.
9:8	<b>ExtMmioMapAddSel: extended MMIO map address select.</b> Read-write. These bits specify the address bits used in [The Extended MMIO Address Base Registers] F1x114_x2, and therefore the granularity of the map registers. It is encoded as: 00b = 0.5 Mbyte granularity.                      10b = 128 Mbyte granularity. 01b = 8 Mbyte granularity.                        11b = Reserved. See F1x114_x2 for details.
7:4	Reserved.
3	<b>Cpu5En: core 5 enable.</b> Revision C and earlier: Reserved. Revision D and later: Read-write. See F0x68[Cpu1En].
2	<b>Cpu4En: core 4 enable.</b> Revision C and earlier: Reserved. Revision D and later: Read-write. See F0x68[Cpu1En].
1	<b>Cpu3En: core 3 enable.</b> Read-write. See F0x68[Cpu1En].
0	<b>Cpu2En: core 2 enable.</b> Read-write. See F0x68[Cpu1En].

### F0x16C Link Global Extended Control Register

Further information about these bits can be found in the Gen3 link specification. BIOS should program this register to the same value in all nodes of a multi-node system.

Bits	Description
31:23	Reserved.





7:6	<p><b>InLnSt: inactive lane state.</b> Read-write. Cold reset: 00b. Specifies the state of inactive lanes of ganged links at Gen3 frequencies as follows:</p> <ul style="list-style-type: none"> <li>00b= Same as warm reset except CAD is logical 0.</li> <li>01b= Same as PHY OFF.</li> <li>10b= Same as operational; CTL and CAD transmit undefined scrambled data.</li> <li>11b= Same as disconnected per <a href="#">F0x[18C:170][LS2En]</a>.</li> </ul> <p>If (InLnSt=11b and <a href="#">F0x[18C:170][LS2En]=0</a>), a link width increase cannot be done after a frequency change, unless it is accompanied by another frequency change or a warm reset.</p> <p>If InLnSt=00b, InLnSt=01b, or (InLnSt=11b and <a href="#">F0x[18C:170][LS2En]=1</a>):</p> <ul style="list-style-type: none"> <li>• For revision DA-C and C3 and later, link width changes are supported via LDTSTOP or warm reset. A width change via LDTSTOP incurs FullT0Time.</li> <li>• For all other revisions, link width changes are only supported via warm reset.</li> </ul> <p>Updates to this field take effect on the next warm reset or LDTSTOP. BIOS should program this field to 01b for revision DA-C processors in the S1g3 package and revision C3 single link processors.</p>
5:0	<p><b>T0Time: training 0 time.</b> Read-write. Cold reset: 3Ah. Specifies the amount of time to spend in training 0 when exiting the disconnected state. See ForceFullT0, <a href="#">F4x1[9C, 94, 8C, 84]_x[530A, 520A][Ls2ExitTime]</a>, and section 2.7.6 [<a href="#">Link LDTSTOP_L Disconnect-Reconnect</a>].</p> <ul style="list-style-type: none"> <li>If T0Time[5:4]=00b, then the time = T0Time[3:0] * 0.1 usec (ranging from 0.0 to 1.5 usec).</li> <li>If T0Time[5:4]=01b, then the time = T0Time[3:0] * 0.5 usec (ranging from 0.0 to 7.5 usec).</li> <li>If T0Time[5:4]=10b, then the time = T0Time[3:0] * 2.0 usec (ranging from 0.0 to 30 usec).</li> <li>If T0Time[5:4]=11b and T0Time[3:0] ranges from 0h to Ah, <ul style="list-style-type: none"> <li>then the time = T0Time[3:0] * 20 usec (ranging from 0.0 to 200 usec).</li> </ul> </li> <li>If T0Time[5:4]=11b and T0Time[3:0] ranges from Bh to Fh, these values are reserved.</li> </ul> <p>BIOS should set T0Time according to the T0 training time requirement for the links' far-side receiver phase recovery time as determined by characterization. The recommended values are 14h if <a href="#">F0x[18C:170][LS2En]=0</a> and 26h if <a href="#">F0x[18C:170][LS2En]=1</a>.</p>

### **F0x[18C:170] Link Extended Control Registers**

These registers provide control for each link. They are mapped to the links as follows:

- F0x170=Link 0, sublink 0. F0x174=Link 1, sublink 0. F0x178=Link 2, sublink 0. F0x17C=Link 3, sublink 0.
- F0x180=Link 0, sublink 1. F0x184=Link 1, sublink 1. F0x188=Link 2, sublink 1. F0x18C=Link 3, sublink 1.

Visibility of these sublink 1 registers is as specified through Ganged (bit 0) of the sublink 0 registers.

If a link is ganged, only the register for sublink 0 of that link is visible and it applies to the whole link. Further information about these bits can be found in the Gen3 link specification.

Bits	Description																																				
31:14	Reserved.																																				
13:12	<p><b>LaneSel: lanes select.</b> Read-write. Cold reset: 00b. This field only exists in the sublink 0 registers; in the sublink 1 registers, these bits are reserved. For ungangged links, they apply to both sublinks. This field specifies how receive (RX) lanes are translated into transmit (TX) lanes for links that are in ILM. The translation varies with link width. Given the RX order specified below, the TX order varies with LaneSel as follows:</p> <table border="0"> <tr> <td><b>Bits</b></td> <td><u>16-bit link</u></td> <td><u>8-bit link</u></td> </tr> <tr> <td></td> <td>RX={CTL1, CAD[15:8], CTL0, CAD[7:0]}</td> <td>RX={CTL0, CAD[7:0]}</td> </tr> <tr> <td>00b</td> <td>Same as RX.</td> <td>Same as RX</td> </tr> <tr> <td>01b</td> <td>TX={CAD[12:8], CTL0, CAD[7:0], CTL1, CAD[15:13]}</td> <td>TX={CAD[6:0], CTL0, CAD[7]}</td> </tr> <tr> <td>10b</td> <td>TX={CTL0, CAD[7:0], CTL1, CAD[15:8]}</td> <td>TX={CAD[4:0], CTL0, CAD[7:5]}</td> </tr> <tr> <td>11b</td> <td>TX={CAD[4:0], CTL1, CAD[15:8], CTL0, CAD[7:5]}</td> <td>TX={CAD[2:0], CTL0, CAD[7:3]}</td> </tr> </table> <table border="0"> <tr> <td><b>Bits</b></td> <td><u>4-bit link</u></td> <td><u>2-bit link</u></td> </tr> <tr> <td></td> <td>RX = {CTL0, CAD[3:0]}</td> <td>RX = {CTL0, CAD[1:0]}</td> </tr> <tr> <td>00b</td> <td>Same as RX.</td> <td>Same as RX.</td> </tr> <tr> <td>01b</td> <td>TX={CAD[3:0], CTL0}</td> <td>TX={CAD[1:0], CTL0}</td> </tr> <tr> <td>10b</td> <td>TX={CAD[2:0], CTL0, CAD[3]}</td> <td>TX={CAD[0], CTL0, CAD[1]}</td> </tr> <tr> <td>11b</td> <td>TX={CAD[1:0], CTL0, CAD[3:2]}</td> <td>Reserved</td> </tr> </table> <p>Note: 01b and 11b are not useful at Gen1 frequencies because the link cannot be trained unless the CTL lanes line up. In BIST mode on 16-bit links, LaneSel[1] selects which sublink is received by the BIST engine. 0=sublink 0, 1=sublink 1; LaneSel[1:0] also causes the receive path of the BIST engine to reverse the translation for 8-bit or smaller links.</p>	<b>Bits</b>	<u>16-bit link</u>	<u>8-bit link</u>		RX={CTL1, CAD[15:8], CTL0, CAD[7:0]}	RX={CTL0, CAD[7:0]}	00b	Same as RX.	Same as RX	01b	TX={CAD[12:8], CTL0, CAD[7:0], CTL1, CAD[15:13]}	TX={CAD[6:0], CTL0, CAD[7]}	10b	TX={CTL0, CAD[7:0], CTL1, CAD[15:8]}	TX={CAD[4:0], CTL0, CAD[7:5]}	11b	TX={CAD[4:0], CTL1, CAD[15:8], CTL0, CAD[7:5]}	TX={CAD[2:0], CTL0, CAD[7:3]}	<b>Bits</b>	<u>4-bit link</u>	<u>2-bit link</u>		RX = {CTL0, CAD[3:0]}	RX = {CTL0, CAD[1:0]}	00b	Same as RX.	Same as RX.	01b	TX={CAD[3:0], CTL0}	TX={CAD[1:0], CTL0}	10b	TX={CAD[2:0], CTL0, CAD[3]}	TX={CAD[0], CTL0, CAD[1]}	11b	TX={CAD[1:0], CTL0, CAD[3:2]}	Reserved
<b>Bits</b>	<u>16-bit link</u>	<u>8-bit link</u>																																			
	RX={CTL1, CAD[15:8], CTL0, CAD[7:0]}	RX={CTL0, CAD[7:0]}																																			
00b	Same as RX.	Same as RX																																			
01b	TX={CAD[12:8], CTL0, CAD[7:0], CTL1, CAD[15:13]}	TX={CAD[6:0], CTL0, CAD[7]}																																			
10b	TX={CTL0, CAD[7:0], CTL1, CAD[15:8]}	TX={CAD[4:0], CTL0, CAD[7:5]}																																			
11b	TX={CAD[4:0], CTL1, CAD[15:8], CTL0, CAD[7:5]}	TX={CAD[2:0], CTL0, CAD[7:3]}																																			
<b>Bits</b>	<u>4-bit link</u>	<u>2-bit link</u>																																			
	RX = {CTL0, CAD[3:0]}	RX = {CTL0, CAD[1:0]}																																			
00b	Same as RX.	Same as RX.																																			
01b	TX={CAD[3:0], CTL0}	TX={CAD[1:0], CTL0}																																			
10b	TX={CAD[2:0], CTL0, CAD[3]}	TX={CAD[0], CTL0, CAD[1]}																																			
11b	TX={CAD[1:0], CTL0, CAD[3:2]}	Reserved																																			
11	<p><b>ILMEn: internal loopback mode (ILM) enable.</b> Read-write. Cold reset: 0. 1=ILM enabled on the next LDTSTOP Disconnect or warm reset. Cleared by hardware upon the subsequent LDTSTOP Disconnect or warm reset. <a href="#">F4x1[9C, 94, 8C, 84]_x[DF, CF][XmtRdPtr and RcvRdPtr]</a> must be 0 (the default) when ILM mode is used.</p>																																				
10	<p><b>BistEn: built-in self test (BIST) enable.</b> Read-write. Cold reset: 0. 1=The link BIST engine is enabled on the next LDTSTOP Disconnect or warm reset. Cleared by hardware upon the subsequent LDTSTOP Disconnect or warm reset.</p>																																				
9	Reserved																																				
8	<p><b>LS2En: LDTSTOP mode 2 enable.</b> Read-write. Cold reset: 0. 0=Use LS1 mode for power reduction when the link is disconnected. 1=Use LS2 mode. BIOS should set this bit for single-link processors and for revision D1 G34 and C32 processors. See <a href="#">F0x16C[ForceFullT0]</a>.</p>																																				
7:4	Reserved.																																				
3	<p><b>ScrambleEn: scrambling enable.</b> Read-write. Cold reset: 0. 1=Scrambling enable. Updates to this bit take effect on the next warm reset or LDTSTOP assertion. Software must clear this bit when transitioning from Gen3 to Gen1 protocol. See <a href="#">Table 32</a>.</p>																																				
2:1	Reserved.																																				
0	<p><b>Ganged.</b> Read-write; read-only 1 if the bit corresponding to the link in <a href="#">F3xE8[UnGangEn]</a> is 0. 0=The link is ungangged; this register is visible for both sublinks. 1=The link is ganged; only the sublink 0 register is visible. This value is initialized after a cold reset, based on the ganging state determined by hardware (see <a href="#">2.7.1.1 [Ganging And Unganging]</a>). Writes to this bit take effect on the next warm reset; reads reflect the last value written (rather than the current state of the link). This bit only exists in the sublink 0 registers.</p>																																				

**F0x1A0 Link Initialization Status Register**

Bits	Description																																				
31	<b>InitStatusValid: initialization status valid.</b> Read-only. 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.																																				
30:24	Reserved.																																				
23:16	<b>IntLnkRoute: internal link routing.</b> Revision C and earlier: Reserved. Revision D and later: Read-only. This field defines an 8 bit vector, with one bit per sublink, indicating whether the links are connected to internal nodes or external nodes. This field is valid for all links regardless of the initialization status if <a href="#">F3xE8[MultiNodeCpu]=1</a> . 1=Internal node. 0=External node. See <a href="#">F0x[E8, C8, A8, 88][Freq[3:0]]</a> and <a href="#">F4x1[9C, 94, 8C, 84]_x[D5, C5]</a> . <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>link 0 sublink 0.</td> <td>4</td> <td>link 0 sublink 1.</td> </tr> <tr> <td>1</td> <td>link 1 sublink 0.</td> <td>5</td> <td>link 1 sublink 1.</td> </tr> <tr> <td>2</td> <td>link 2 sublink 0.</td> <td>6</td> <td>link 2 sublink 1.</td> </tr> <tr> <td>3</td> <td>link 3 sublink 0.</td> <td>7</td> <td>link 3 sublink 1.</td> </tr> </tbody> </table>	Bit	Description	Bit	Description	0	link 0 sublink 0.	4	link 0 sublink 1.	1	link 1 sublink 0.	5	link 1 sublink 1.	2	link 2 sublink 0.	6	link 2 sublink 1.	3	link 3 sublink 0.	7	link 3 sublink 1.																
Bit	Description	Bit	Description																																		
0	link 0 sublink 0.	4	link 0 sublink 1.																																		
1	link 1 sublink 0.	5	link 1 sublink 1.																																		
2	link 2 sublink 0.	6	link 2 sublink 1.																																		
3	link 3 sublink 0.	7	link 3 sublink 1.																																		
15:0	<b>NC and InitComplete.</b> Read-only. These bits provide duplicate versions of status bits <a href="#">F0x[F8, D8, B8, 98][NC and InitComplete]</a> and <a href="#">F4x[F8, D8, B8, 98][NC and InitComplete]</a> as follows: <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>link 0 sublink 0 InitComplete.</td> <td>8</td> <td>link 0 sublink 1 InitComplete.</td> </tr> <tr> <td>1</td> <td>link 0 sublink 0 NC.</td> <td>9</td> <td>link 0 sublink 1 NC.</td> </tr> <tr> <td>2</td> <td>link 1 sublink 0 InitComplete.</td> <td>10</td> <td>link 1 sublink 1 InitComplete.</td> </tr> <tr> <td>3</td> <td>link 1 sublink 0 NC.</td> <td>11</td> <td>link 1 sublink 1 NC.</td> </tr> <tr> <td>4</td> <td>link 2 sublink 0 InitComplete.</td> <td>12</td> <td>link 2 sublink 1 InitComplete.</td> </tr> <tr> <td>5</td> <td>link 2 sublink 0 NC.</td> <td>13</td> <td>link 2 sublink 1 NC.</td> </tr> <tr> <td>6</td> <td>link 3 sublink 0 InitComplete.</td> <td>14</td> <td>link 3 sublink 1 InitComplete.</td> </tr> <tr> <td>7</td> <td>link 3 sublink 0 NC.</td> <td>15</td> <td>link 3 sublink 1 NC.</td> </tr> </tbody> </table>	Bit	Description	Bit	Description	0	link 0 sublink 0 InitComplete.	8	link 0 sublink 1 InitComplete.	1	link 0 sublink 0 NC.	9	link 0 sublink 1 NC.	2	link 1 sublink 0 InitComplete.	10	link 1 sublink 1 InitComplete.	3	link 1 sublink 0 NC.	11	link 1 sublink 1 NC.	4	link 2 sublink 0 InitComplete.	12	link 2 sublink 1 InitComplete.	5	link 2 sublink 0 NC.	13	link 2 sublink 1 NC.	6	link 3 sublink 0 InitComplete.	14	link 3 sublink 1 InitComplete.	7	link 3 sublink 0 NC.	15	link 3 sublink 1 NC.
Bit	Description	Bit	Description																																		
0	link 0 sublink 0 InitComplete.	8	link 0 sublink 1 InitComplete.																																		
1	link 0 sublink 0 NC.	9	link 0 sublink 1 NC.																																		
2	link 1 sublink 0 InitComplete.	10	link 1 sublink 1 InitComplete.																																		
3	link 1 sublink 0 NC.	11	link 1 sublink 1 NC.																																		
4	link 2 sublink 0 InitComplete.	12	link 2 sublink 1 InitComplete.																																		
5	link 2 sublink 0 NC.	13	link 2 sublink 1 NC.																																		
6	link 3 sublink 0 InitComplete.	14	link 3 sublink 1 InitComplete.																																		
7	link 3 sublink 0 NC.	15	link 3 sublink 1 NC.																																		

**F0x1E0 Coherent Link Pair Traffic Distribution Register**

Revision C and earlier: Reserved. Revision D and later: Reset: 0000 0000h. See [2.6.4.2.4 \[Link Traffic Distribution\]](#) for details about link traffic distribution.

31:29	<b>AltSel3: alternate select 3.</b> See <a href="#">AltSel0</a> .
28:26	<b>MasterSel3: master select 3.</b> See <a href="#">MasterSel0</a> .
25	<b>Asym3: asymmetric 3.</b> See <a href="#">Asym0</a> .
24	<b>DistEn3: distribution enable 3.</b> See <a href="#">DistEn0</a> .
23:21	<b>AltSel2: alternate select 2.</b> See <a href="#">AltSel0</a> .
20:18	<b>MasterSel2: master select 2.</b> See <a href="#">MasterSel0</a> .
17	<b>Asym2: asymmetric 2.</b> See <a href="#">Asym0</a> .
16	<b>DistEn2: distribution enable 2.</b> See <a href="#">DistEn0</a> .
15:13	<b>AltSel1: alternate select 1.</b> See <a href="#">AltSel0</a> .
12:10	<b>MasterSel1: master select 1.</b> See <a href="#">MasterSel0</a> .
9	<b>Asym1: asymmetric 1.</b> See <a href="#">Asym0</a> .
8	<b>DistEn1: distribution enable 1.</b> See <a href="#">DistEn0</a> .

7:5	<b>AltSel0: alternate select 0.</b> Read-write. 3-bit select for alternate link of pair 0. See MasterSel0 for the encodings.
4:2	<b>MasterSel0: master select 0.</b> Read-write. 3-bit select for master link of pair 0. 000b=link 0 sublink 0.                      100b=link 0 sublink 1. 001b=link 1 sublink 0.                      101b=link 1 sublink 1. 010b=link 2 sublink 0.                      110b=link 2 sublink 1. 011b=link 3 sublink 0.                      111b=link 3 sublink 1.
1	<b>Asym0: asymmetric 0.</b> Read-write. 1=Link pair 0 is asymmetric.
0	<b>DistEn0: distribution enable 0.</b> Read-write. 1=Enables traffic distribution for link pair 0.

### 3.4 Function 1 Address Map Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention. See 2.11 [Configuration Space] for details about how to access this space.

#### F1x00 Device/Vendor ID Register

Reset: 1201 1022h.

Bits	Description
31:16	<b>DeviceID: device ID.</b> Read-only.
15:0	<b>VendorID: vendor ID.</b> Read-only.

#### F1x08 Class Code/Revision ID Register

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. Processor revision. 00h=A0.

#### F1x0C Header Type Register

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.

#### F1x[1, 0][7C:40] DRAM Base/Limit Registers

These registers specify the destination node of DRAM address ranges. The following 8 sets of registers are specified:

<u>Base Address</u>	<u>Limit Address</u>	<u>Base Address</u>	<u>Limit Address</u>
F1x040, F1x140	F1x044, F1x144	F1x060, F1x160	F1x064, F1x164
F1x048, F1x148	F1x04C, F1x14C	F1x068, F1x168	F1x06C, F1x16C
F1x050, F1x150	F1x054, F1x154	F1x070, F1x170	F1x074, F1x174
F1x058, F1x158	F1x05C, F1x15C	F1x078, F1x178	F1x07C, F1x17C

F1x0XX registers provide the low address bits and F1x1XX registers provide the high address bits. Transaction addresses that are within the specified base/limit range are routed to the DstNode. See [The Northbridge

### Routing] 2.6.4.

DRAM mapping rules:

- Transaction addresses are within the defined range if:  
 $\{\text{DramBase}[47:24], 00\_0000\text{h}\} \leq \text{address}[47:0] \leq \{\text{DramLimit}[47:24], \text{FF\_FFFFh}\}$ .
- DRAM regions must not overlap each other.
- Accesses to addresses that map to both DRAM, as specified by  $\text{F1x}[1, 0][7\text{C}:40]$ , and MMIO, as specified by  $\text{F1x}[BC:80]$ , 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 [\[The DRAM Base/Limit Registers\] 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 [2.6.4.1.1 \[DRAM and MMIO Memory Space\]](#).

**Hoisting.** When memory hoisting is enabled in a node (via  $\text{F1xF0}[\text{DramHoleValid}]$ ),  $\text{F1x}[1, 0][7\text{C}:40][\text{DramLimit}]$  should be set up to account for the memory hoisted above the hole. I.e.,  $\text{F1x}[1, 0][7\text{C}:40][\text{DramLimit}]$  should be set to  $\text{F1x}[1, 0][7\text{C}:40][\text{DramBase}]$  plus the size of the amount of memory owned by the node plus the hole size (4G minus  $\text{F1xF0}[\text{DramHoleBase}]$ ). See [2.8.11 \[Memory Hoisting\]](#) for more information about memory hoisting.

**Node interleave.** DRAM may be mapped as continuous regions for each node or it may be interleaved between nodes. See [2.8.10.2 \[Node Interleaving\]](#) for details.

#### **F1x[78, 70, 68, 60, 58, 50, 48, 40] DRAM Base Address Registers**

Bits	Description
31:16	<b>DramBase[39:24]: DRAM base address register bits[39:24].</b> Read-write. Reset: 0000h.
15:11	Reserved
10:8	<b>IntlvEn[2:0]: interleave enable.</b> Read-write. Reset: 0h. This field enables interleaving on a 4-Kbyte boundary between memory on different nodes. The bits are encoded as follows: 000b = No interleave 001b = Interleave on A[12] (2 nodes) 011b = Interleave on A[12] and A[13] (4 nodes) 111b = Interleave on A[12], A[13], and A[14] (8 nodes) All other values are reserved. The value of this field is required to match <a href="#">F1x124[DramIntlvEn]</a> .
7:2	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.

#### **F1x[178, 170, 168, 160, 158, 150, 148, 140] DRAM Base Address High Registers**

Bits	Description
31:8	Reserved.
7:0	<b>DramBase[47:40]: DRAM base address register bits[47:40].</b> Read-write. Reset: 0.

**F1x[7C, 74, 6C, 64, 5C, 54, 4C, 44] DRAM Limit Address Registers**

Bits	Description
31:16	<b>DramLimit[39:24]: DRAM limit address register bits[39:24].</b> Read-write. Reset: X, except in F1x44 reset: FFFFh.
15:11	Reserved.
10:8	<b>IntlvSel: interleave select.</b> Read-write. Reset: X, except in F1x44 reset: 000b. This field specifies the values of address bits A[14:12] to use with the Interleave Enable field (IntlvEn[2:0]) to determine which 4-Kbyte blocks are routed to this region. IntlvSel[0] corresponds to A[12]; IntlvSel[1] corresponds to A[13]; IntlvSel[2] corresponds to A[14].
7:3	Reserved.
2:0	<b>DstNode: destination Node ID.</b> Read-write. Reset: X, except in F1x44 reset: 0h. This field specifies the node that a packet is routed to if it is within the address range.

**F1x[17C, 174, 16C, 164, 15C, 154, 14C, 144] DRAM Limit Address High Registers**

Bits	Description
31:8	Reserved.
7:0	<b>DramLimit[47:40]: DRAM limit address register bits[47:40].</b> Read-write. Reset: 0.

**F1x[BC:80] Memory Mapped IO Base/Limit Registers**

These registers specify the mapping from memory addresses to the corresponding node and IO link for MMIO transactions. Address ranges are specified by 8 sets of base/limit registers. The first set is F1x80 and F1x84, the second set is F1x88 and F1x8C, and so forth. Transaction addresses that are within the specified base/limit range are routed to the node specified by DstNode and the link specified by DstLink. See [\[The Northbridge Routing\] 2.6.4](#).

MMIO mapping rules:

- Transaction addresses are within the defined range if:  
 $\{00h, MMIOBase[39:16], 0000h\} \leq \text{address}[47:0] \leq \{00h, MMIOLimit[39:16], FFFFh\}$ .
- MMIO regions must not overlap each other.
- Accesses to addresses that map to both DRAM, as specified by [F1x\[1, 0\]\[7C:40\]](#), and MMIO, as specified by [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 [\[The Memory Mapped IO Base/Limit Registers\] F1x\[BC:80\]](#), only.
- The appropriate RE or WE bit(s) must be set. When initializing a base/limit pair, the BIOS must write the limit address 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 [2.6.4.1.1 \[DRAM and MMIO Memory Space\]](#).

**F1x[B8, B0, A8, A0, 98, 90, 88, 80] MMIO Base Address Registers**

Bits	Description
31:8	<b>MMIOBase[39:16]: MMIO base address register bits[39:16].</b> Read-write. Reset: X.
7:4	Reserved.



3	<b>Lock.</b> Read-write. Reset: X. 1=The MMIO base/limit address register pair is read-only (including this bit) unless both WE and RE bits are 0.
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.

### F1x[BC, B4, AC, A4, 9C, 94, 8C, 84] MMIO Limit Address Registers

Bits	Description
31:8	<b>MMIOLimit[39:16]: MMIO limit address register bits[39:16].</b> Read-write. Reset: X.
7	<b>NP: non-posted.</b> Read-write. Reset: X. 1=CPU write requests to this MMIO range are passed through the non-posted channel. This may be used to force writes to be non-posted for MMIO regions which map to the legacy ISA/LPC bus, or in conjunction with <a href="#">[The Link Transaction Control Register] F0x68[DsNpReqLmt]</a> 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 <i>HyperTransport™ IO Link Specification</i> 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).  Note: if two MMIO ranges target the same IO device and the NP bit is set differently in both ranges, unexpected transaction ordering effects are possible. In particular, using PCI- and IO-link-defined producer-consumer semantics, if a producer (e.g., the processor) writes data using a non-posted MMIO range followed by a flag to a posted MMIO range, then it is possible for the device to see the flag updated before the data is updated.
6	<b>DstSubLink: destination sublink.</b> Read-write. Reset: X. When a link is ungangled, this bit specifies the destination sublink of the link specified by F1x[BC:80][DstLink]. 0=The destination link is sublink 0. 1=The destination link is sublink 1. If the link is gangled, then this bit must be low.
5:4	<b>DstLink: destination link ID.</b> Read-write. Reset: X. For transactions within the this MMIO range, this field specifies the destination IO link number of the destination node. 00b = Link 0            10b = Link 2 01b = Link 1            11b = Link 3
3	Reserved.
2:0	<b>DstNode: destination node ID bits.</b> Read-write. Reset: X. For transactions within the this MMIO range, this field specifies the destination node ID.

### F1x[DC:C0] IO-Space Base/Limit Registers

These registers 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 4 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 [\[The Northbridge Routing\] 2.6.4](#).

IO mapping rules:

- IO-space transaction addresses are within the defined range if:  
{IOBase[24:12], 000h} <= address <= {IOLimit[24:12], FFFh} and as specified by the IE bit; or  
if the address is in the range specified by the VE bits.
- IO regions must not overlap each other.

- The appropriate RE or WE bit(s) must be set.
- See 2.6.4.1.2 [IO Space].

### F1x[D8, D0, C8, C0] IO-Space Base Address Registers

Bits	Description
31:25	Reserved.
24:12	<b>IOBase[24:12]: IO base address register bits[24:12].</b> Read-write. Reset: X.
11:6	Reserved.
5	<b>IE: ISA enable.</b> Read-write. Reset: X. 1=The IO-space address window is limited to the first 256 bytes of each 1K byte block specified; this only applies to the first 64K bytes of IO space. 0=The PCI IO window is not limited in this way.
4	<b>VE: VGA enable.</b> Read-write. Reset: X. 1=Include IO-space transactions targeting the VGA-compatible address space within the IO-space window of this base/limit pair. These include IO accesses in which address bits[9:0] range from 3B0h to 3BBh or 3C0h to 3DFh (address bits[15:10] are not decoded); this only applies to the first 64K of IO space; i.e., address bits[24:16] must be low). 0=IO-space transactions targeting VGA-compatible address ranges are not added to the IO-space window. This bit should only ever be set in one register. Note: 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 [The VGA Enable Register] F1xF4. Note, when F1xF4[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.

### F1x[DC, D4, CC, C4] IO-Space Limit Address Registers

Bits	Description
31:25	Reserved.
24:12	<b>IOLimit[24:12]: IO limit address register bits[24:12].</b> Read-write. Reset: X.
11:7	Reserved.
6	<b>DstSubLink: destination sublink.</b> Read-write. Reset: X. When a link is ungang, 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 gang, then this bit must be low.
5:4	<b>DstLink: destination link ID.</b> Read-write. Reset: X. For transactions within the this IO-space range, this field specifies the destination IO link number of the destination node. 00b = Link 0            10b = Link 2 01b = Link 1            11b = Link 3
3	Reserved.
2:0	<b>DstNode: destination node ID bits.</b> Read-write. Reset: X. For transactions within the this IO-space range, this field specifies the destination node ID.

### F1x[EC:E0] Configuration Map Registers

These registers specify the mapping from configuration address to the corresponding node and IO link. Configuration address ranges are specified by 4 pairs of base/limit registers. The first is F1xE0, the second is F1xE4, 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 [The Northbridge Routing] 2.6.4.

Configuration space mapping rules:

- Configuration addresses (to “BusNo” and “Device” as specified by [The IO-Space Configuration Address Register] IOCF8 in the case of IO accesses or [The Configuration Space] 2.11 in the case of MMIO accesses) are within the defined range if:  
 $(\{BusNumBase[7:0]\} \leq BusNo \leq \{BusNumLimit[7:0]\}) \& (DevCmpEn==0);$  or  
 $(\{BusNumBase[4:0]\} \leq Device \leq \{BusNumLimit[4:0]\}) \& (DevCmpEn==1) \& (BusNo == 00h).$
- Configuration regions must not overlap each other.
- The appropriate RE or WE bit(s) must be set.
- See 2.6.4.1.3 [Configuration Space].

Bits	Description
31:24	<b>BusNumLimit[7:0]: bus number limit bits[7:0].</b> Read-write. Reset: X.
23:16	<b>BusNumBase[7:0]: bus number base bits[7:0].</b> Read-write. Reset: X.
15:11	Reserved.
10	<b>DstSubLink: destination sublink.</b> Read-write. Reset: X. When a link is ungangled, this bit specifies the destination sublink of the link specified by <b>F1x[EC:E0][DstLink]</b> . 0=The destination link is sublink 0. 1=The destination link is sublink 1. If the link is gangled, then this bit must be low.
9:8	<b>DstLink: destination link ID.</b> Read-write. Reset: X. For transactions within the this configuration-space range, this field specifies the destination IO link number of the destination node. 00b = Link 0 01b = Link 1 10b = Link 2 11b = Link 3
7	Reserved.
6:4	<b>DstNode: destination node ID bits.</b> Read-write. Reset: X. For transactions within the this configuration-space range, this field specifies the destination node ID.
3	Reserved.
2	<b>DevCmpEn: device number compare mode enable.</b> Read-write. Reset: X. 1=A device number range rather than a bus number range is used to specify the configuration-space window (see above). This is used to enable multiple IO links to be configured as Bus 0.
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.

### F1xF0 DRAM Hole Address Register

Reset: 0000 0000h.

Bits	Description
31:24	<b>DramHoleBase[31:24].</b> DRAM hole base address. Read-write. This specifies the base address of the IO hole, below the 4G address level, that is used in memory hoisting. Normally, <b>DramHoleBase</b> >= <b>MSRC001_001A[TOM[31:24]]</b> . See 2.8.11 [Memory Hoisting] for additional programming information.
23:16	Reserved.

15:7	<ul style="list-style-type: none"> <li>• <b>DramHoleOffset[31:23]: DRAM hole offset address.</b> Read-write. When memory hoisting is enabled, this value is subtracted from the physical address of certain transactions before being passed to the DCT. See 2.8.11 [Memory Hoisting] for additional programming information.</li> </ul>
6:2	Reserved.
1	<b>DramMemHoistValid.</b> Read-write. 1=Memory hoisting is enabled in one or more nodes of the coherent fabric. This bit should be set in all nodes of the coherent fabric if memory hoisting is employed by any of them. See 2.8.11 [Memory Hoisting] for additional programming information.
0	<b>DramHoleValid.</b> Read-write. 1=Memory hoisting is enabled in the node. 0=Memory hoisting is not enabled. This bit should be set in the node(s) that own the DRAM address space that is hoisted above the 4GB address level. If node interleaving is employed, then this should be set in all nodes. See 2.8.11 [Memory Hoisting] for additional programming information.

### F1xF4 VGA Enable Register

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 ungangd, this bit specifies the destination sublink of the link specified by F1xF4[DstLink]. 0=The destination link is sublink 0. 1=The destination link is sublink 1. If the link is gangd, then this bit must be low.
13:12	<b>DstLink: destination link ID.</b> Read-write. For transactions within the F1xF4[VE]-defined ranges, this field specifies the destination IO link number of the destination node. 00b = Link 0 01b = Link 1 10b = Link 2 11b = Link 3
11:7	Reserved.
6:4	<b>DstNode[2:0]: destination node ID.</b> Read-write. For transactions within the F1xF4[VE]-defined range, this field specifies the destination node ID.
3	<b>Lock.</b> Read-write. Reset: 0. 1=All the bits in this register (F1xF4) are read-only (including this bit).
2	Reserved.
1	<b>NP: non-posted.</b> Read-write. 1=CPU write requests to the F1xF4[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 64K 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. Note, when this bit is set, the state of F1x[DC:C0][VE] is ignored.

### F1x110 Extended Address Map Control Register

This register provides the index to several extended address map control registers. In order to access these registers, (1) AddrMapType and Index are written into this register; (2) read-write access to the register is accomplished through [The Extended Address Map Data Port] F1x114. The extended address map registers are

disabled when [The Link Transaction Control Register] F0x68[CHtExtAddrEn] = 0b; when disabled the extended address maps are not checked for routing packets. F1x110 must not be accessed if F0x68[CHtExtAddrEn]=0b.

Before reading F1x114\_x2 or F1x114\_x3 software must initialize the registers or NB Array MCA errors may occur. If F0x68[CHtExtAddrEn]=0, BIOS should initialize index 0h of F1x114\_x2 and F1x114\_x3. If F0x68[CHtExtAddrEn]=1, BIOS should initialize all indices of F1x114\_x2 and F1x114\_x3.

Bits	Description
31	Reserved.
30:28	<b>AddrMapType</b> . Read-write. Specifies the type of address map being accessed as follows: 00xb = Reserved 010b = [The Extended MMIO Address Base Registers] F1x114_x2 011b = [The Extended MMIO Address Mask Registers] F1x114_x3 1xxb = Reserved
27:4	Reserved.
3:0	<b>Index</b> . Read-write. This function varies based on the AddrMapType register accessed.

### F1x114 Extended Address Map Data Port

See F1x110 for details about this port.

### F1x114\_x2 Extended MMIO Address Base Registers

See F1x110 for information about accessing this set of registers. The extended MMIO address map is a 16 entry table with fully associative lookup. Each entry is accessed through F1x114\_x2 and F1x114\_x3, with F1x110[Index] specifying the entry number. An incoming transaction of address Addr[47:0] is determined to be within the range specified by an entry if the following is true, as a function of F0x168[ExtMmioMapAddSel]:

#### F0x168[ExtMmioMapAddSel]:Equation

00b: (Addr[39:19] | MmioMapMask[20:0] == MmioMapBase[20:0] | MmioMapMask[20:0]) & (Addr[47:40] == 00h)  
01b: (Addr[43:23] | MmioMapMask[20:0] == MmioMapBase[20:0] | MmioMapMask[20:0]) & (Addr[47:44] == 0h)  
10b: (Addr[47:27] | MmioMapMask[20:0] == MmioMapBase[20:0] | MmioMapMask[20:0])  
11b: Reserved.

Accesses within the range specified by an entry are routed to the node specified by MmioDstNode.

It is the responsibility of software to ensure each address hits only 1 entry in the MMIO map. Hits to multiple entries result in undefined behavior. Note the precedence of defined memory ranges specified by section 2.6.4.1.1 [DRAM and MMIO Memory Space].

Note that the MMIO base and mask entries are written into the address map together by the hardware only when the mask is written by software. As a result, the base (F1x114\_x2) must be written by software before the mask (F1x114\_x3). On a read, the mask must be read before the base and the hardware read of the map registers occurs when the mask is read by software. Also, writes to the data/mask registers are stored differently than they are written, such that the value read back may be different than what is written as follows:

<u>Write base bit</u>	<u>Write mask bit</u>	<u>Read base bit</u>	<u>Read mask bit</u>	<u>Notes</u>
0	0	0	1	Base bit is 0, unmasked
1	0	1	0	Base bit is 1, unmasked
0 or 1	1	1	1	Bit is masked

All CPU write requests that are routed through these registers are routed in the posted channel. IO link write requests that are routed through these registers use channel indicated in the source request.

Bits	Description
31:29	Reserved.
28:8	<b>MmioMapBase[20:0]</b> . Read-write. Reset: X.
7	Reserved.
6	<b>MmioDstThisNode</b> . Read-write. Reset: X. 1=The destination is the local node. See MmioDstNode.
5:3	Reserved.
2:0	<b>MmioDstNode</b> . Read-write. Reset: X. Specifies the destination node or link of the MMIO access. If MmioDstThisNode=1, MmioDstNode[1:0] contains the destination link number and MmioDstNode[2] contains the destination sublink (if the link is ungangled).

### **F1x114\_x3 Extended MMIO Address Mask Registers**

See [F1x114\\_x2](#) for details.

Bits	Description
31:29	Reserved.
28:8	<b>MmioMapMask[20:0]</b> . Read-write. Reset: X. 1=Address bit is a don't care.
7:1	Reserved.
0	<b>MmioMapEn</b> . Read-write. Reset: 0. 1=This entry is enabled.

### **F1x120 DRAM Base System Address Register**

[F1x120](#) and [F1x124](#) 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:

{DramBaseAddr[47:27], 000\_0000h} <= Addr[47:0] <= {DramLimitAddr[47:27], 7FF\_FFFFh};

DRAM accesses to the local node that are outside of this range are master aborted. This range is also used to specify the range of DRAM covered by the scrubber (see [F3x58](#) and [F3x5C](#)).

DRAM may be mapped as continuous regions for each node or it may be interleaved between nodes. If node interleaving is not invoked, as specified by DramIntlvEn, then the address of the DRAM transaction is normalized before passing it to the DCTs by subtracting DramBaseAddr.

If node interleaving is invoked, then DramBaseAddr should be zero in all the nodes and DramLimitAddr should be the top of memory in all nodes. Based on the value of DramIntlvEn, the normalized address to the DCTs is modified to remove the affected address bits between A[17:12]; e.g., if 8-node interleave is invoked, then DramIntlvEn is set to 111b and the normalized address to the DCTs removes A[14:12] to become {A[47:15], A[11:0]}. See [2.8.10.2 \[Node Interleaving\]](#) for more details.

Bits	Description
31:24	Reserved.
23:21	<b>DramIntlvSel: interleave select.</b> Read-write. Reset: 0. This field specifies the values of address bits A[14:12] that are routed to the local node when node interleaving is enabled. IntlvSel[0] corresponds to A[12]; IntlvSel[1] corresponds to A[13]; IntlvSel[2] corresponds to A[14].
20:0	<b>DramBaseAddr[47:27]: dram base address.</b> Read-write. Reset: 0.

### F1x124 DRAM Limit System Address Register

See [F1x120](#).

Bits	Description
31:21	Reserved.
23:21	<b>DramIntlvEn[2:0]: dram interleave enable.</b> Read-write. Reset: 0. This field specifies interleaving on a 4-Kbyte boundary between DRAM on different nodes. The bits are encoded as follows: 000b = No interleave 001b = Interleave on A[12] (2 nodes) 011b = Interleave on A[12] and A[13] (4 nodes) 111b = Interleave on A[12], A[13], and A[14] (8 nodes) All other values are reserved. The value of this field is required to match <a href="#">F1x[1, 0][7C:40][IntlvEn]</a> .
20:0	<b>DramLimitAddr[47:27]: dram limit address.</b> Read-write. Reset: 1F_FFFFh.

### 3.5 Function 2 DRAM Controller Registers

See [3.1 \[Register Descriptions and Mnemonics\]](#) for a description of the register naming convention. See [2.11 \[Configuration Space\]](#) for details about how to access this space.

#### F2x00 Device/Vendor ID Register

Reset: 1202 1022h.

Bits	Description
31:16	<b>DeviceID: device ID.</b> Read-only.
15:0	<b>VendorID: vendor ID.</b> Read-only.

#### F2x08 Class Code/Revision ID Register

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.

#### F2x0C Header Type Register

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.



## F2x[1, 0][5C:40] DRAM CS Base Address Registers

Reset: 0000 0000h. See 2.8.1 [DCT Configuration Registers] for general programming information about DCT configuration registers.

These registers along with [The DRAM CS Mask Registers] F2x[1, 0][6C:60], translate DRAM request addresses (to a DRAM controller) into DRAM chip selects. Supported DIMM sizes are specified in [The DRAM Bank Address Mapping Register] F2x[1, 0]80. For more information on the DRAM controllers, see 2.8 [DRAM Controllers (DCTs)].

The processor logically supports the following number of DIMMs in the following packages:

**Table 83: DIMM support per package**

Package	Number of DIMMs per channel			
	Registered	4-Rank registered	Unbuffered	SO-DIMMs
Fr2(1207)	4	2	0	0
G34	2 or 3	1 or 2	0	0
AM2r2 and AM3	0	0	2	1
S1g3 or S1g4	0	0	2	1

For each chip select, there is a DRAM CS Base Address register. For every two chip selects there is a DRAM CS Mask Register. These are associated with logical DIMM numbers, CKE, and ODT signals as follows:

**Table 84: Logical DIMM, Chip Select, CKE, ODT, and Register Mapping**

Base Address Registers	Mask Register	Logical DIMM <sup>1</sup>		Chip Select Pad <sup>3</sup>	M[B, A]_CKE[x]	ODT Pad <sup>3</sup>
		N	R4			
F2x[1, 0]40	F2x[1, 0]60	0	0	MEMCS[1:0]_L[0]	0	MEMODT[1:0][0]
F2x[1, 0]44				MEMCS[1:0]_L[1]	1	MEMODT[1:0][1] <sup>4</sup>
F2x[1, 0]48	F2x[1, 0]64	1	1	MEMCS[1:0]_L[2]	0	MEMODT[1:0][2] <sup>4</sup> MEMODT[1:0][1] <sup>4</sup>
F2x[1, 0]4C				MEMCS[1:0]_L[3]	1	MEMODT[1:0][3] <sup>4</sup>
F2x[1, 0]50	F2x[1, 0]68	2 <sup>2</sup>	0	MEMCS[1:0]_L[4]	0	MEMODT[1:0][2]
F2x[1, 0]54				MEMCS[1:0]_L[5]	1	-
F2x[1, 0]58	F2x[1, 0]6C	3 <sup>2</sup>	1	MEMCS[1:0]_L[6]	0	MEMODT[1:0][3]
F2x[1, 0]5C				MEMCS[1:0]_L[7]	1	-

1. N=Normal.

R4=Four-rank registered DIMM only (F2x[1, 0]94[FourRankRDimm]=1).

2. Logical DIMM numbers 2 and 3 are not supported in all packages.

3. See section 2.8.2 [DDR Pad to Processor Pin Mapping].

4. If (F2x[1, 0]94[Ddr3Mode]=1 && F2x[1, 0]90[UnbuffDimm]==1) || (F2x[1, 0]94[PowerDownMode]==1 && F2x[1, 0]94[PowerDownEn]==1) then the first specified pad is used for ODT to the second rank of a dual rank DIMM; otherwise either the second specified pad is used, or the termination for that rank is strapped to the appropriate voltage rail on the platform.

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 2.6.1 [North-bridge (NB) Architecture].

Each base address register specifies the starting normalized address of the block of memory associated with the chip select. Each mask register specifies the additional address bits that are consumed by the block of memory associated with the chip selects. If both chip selects of a logical 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 2.8.10.1 [Chip Select Interleaving] for more information. The hardware supports the use of lower-order address bits to interleave chip selects if (1) 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.

System BIOS is required to assign the largest DIMM chip-select range to the lowest normalized address of the DRAM controller. As addresses increase, the chip-select size is required to remain constant or decrease. This is necessary to keep DIMM chip-select banks on aligned address boundaries, regardless as to the amount of address space covered by each chip select.

For each normalized address for requests that enters a DRAM controller, a ChipSelect[i] is asserted if:

```
CSEnable[i] &
( { (InputAddr[36:27] & ~AddrMask[i][36:27]),
    (InputAddr[21:13] & ~AddrMask[i][21:13]) } ==
  { (BaseAddr[i][36:27] & ~AddrMask[i][36:27]),
    (BaseAddr[i][21:13] & ~AddrMask[i][21:13]) } );
```

Bits	Description
31:29	Reserved.
28:19	<b>BaseAddr[36:27]: normalized physical base address bits [36:27].</b> Read-write.
18:14	Reserved.
13:5	<b>BaseAddr[21:13]: normalized physical base address bits [21:13].</b> Read-write.
4	Reserved.
3	<b>OnDimmMirror: on-DIMM mirroring (ODM) enabled.</b> Read-write. 1=Address and bank bits are swapped for this chip select during DRAM initialization (MRS commands) in order to account for swapped routing on the DIMM. This is expected to be set appropriately for the odd numbered rank of each unbuffered DIMM when F2x[1, 0]90[UnbuffDimm]=1 and F2x[1, 0]94[Ddr3Mode]=1; it is not expected to be set when connected to SO-DIMM or micro-DIMMs. The bits that are swapped when this is enabled are: <ul style="list-style-type: none"> <li>• M[B, A]_BANK[0] and M[B, A]_BANK[1].</li> <li>• M[B, A]_ADD[3] and M[B, A]_ADD[4].</li> <li>• M[B, A]_ADD[5] and M[B, A]_ADD[6].</li> <li>• M[B, A]_ADD[7] and M[B, A]_ADD[8].</li> </ul>
2	<b>TestFail: memory test failed.</b> Read-write. Set by BIOS to indicate that a rank is present but has failed training or a consistency test. BIOS should treat CSEnable=1 and TestFail=1 as mutually exclusive.

1	<b>Spare: spare rank.</b> Read-write. This bit identifies the chip select associated with the spare rank. See <a href="#">2.8.12 [On-Line Spare]</a> .
0	<b>CSEnable: chip select enable.</b> Read-write.

### F2x[1, 0][6C:60] DRAM CS Mask Registers

Reset: 0000 0000h. See [2.8.1 \[DCT Configuration Registers\]](#) for general programming information about DCT configuration registers. See [F2x\[1, 0\]\[5C:40\]](#) for information about this register.

Bits	Description
31:29	Reserved.
28:19	<b>AddrMask[36:27]: normalized physical address mask bits [36:27].</b> Read-write.
18:14	Reserved.
13:5	<b>AddrMask[21:13]: normalized physical address mask bits [21:13].</b> Read-write.
4:0	Reserved.

### F2x[1, 0]78 DRAM Control Register

Reset: 0000 0006h. See [2.8.1 \[DCT Configuration Registers\]](#) for general programming information about DCT configuration registers.

Bits	Description
31:22	<b>MaxRdLatency: maximum read latency.</b> Read-write. This field should be programmed by the system BIOS to specify the maximum round-trip latency in the system from the processor to the DRAM devices and back. The time is specified in NB clocks at NB P-state 0 and includes the asynchronous and synchronous latencies. See <a href="#">2.8.9.9.5 [Calculating MaxRdLatency]</a> .
21:20	Reserved.
19	<b>EarlyArbEn: early arbitration enable.</b> Revision B: Read-write. 1=The DCT optimizes the arbitration phases to improve performance on back-to-back DRAM reads under certain conditions. BIOS should set this bit whenever the NCLK to MEMCLK ratio is between 4.5:1 and 3:1 inclusive. 0=The DCT arbitrates normally. Revision C and later: Read-write. 1=The DCT optimizes the arbitration phases to improve performance on all DRAM commands and BIOS should always set this bit.
18	<b>DqsRcvEnTrain: DQS receiver enable training mode.</b> Read-write. 1=Enable DQS receiver enable training mode. 0 = Normal DQS receiver enable operation.
17:16	Reserved.
15	<b>ChSetupSync: channel setup synchronize.</b> Read-write. 1=To accommodate different channel address and command settings, the DRAM controller internally phase aligns the memory clocks between the two channels regardless of the configured coarse settings in <a href="#">F2x[1, 0]9C_x04</a> . 0=The DRAM controller derives its setup information from <a href="#">F2x[1, 0]9C_x04</a> . Note: BIOS must set this bit to synchronize setup information between the two channels when both of the following conditions are true: <ul style="list-style-type: none"> <li>• The DCTs are in ganged mode.</li> <li>• <a href="#">F2x[1, 0]9C_x04[AddrCmdSetup, CsOdtSetup, CkeSetup]</a> setups for one DCT are all 0s and at least one of the setups, <a href="#">F2x[1, 0]9C_x04[AddrCmdSetup, CsOdtSetup, CkeSetup]</a>, of the other controller is 1.</li> </ul>
15:14	Reserved.

13:12	<b>Trdrd[3:2]: read to read timing.</b> Read-write. This field along with <a href="#">F2x[1, 0]8C[Trdrd[1:0]]</a> combine to specify a 4-bit value, <a href="#">Trdrd[3:0]</a> , when <a href="#">F2x[1, 0]94[Ddr3Mode]=1</a> . See <a href="#">F2x[1, 0]8C[Trdrd[1:0]]</a> .														
11:10	<b>Twrrwr[3:2]: write to write timing.</b> Read-write. This field along with <a href="#">F2x[1, 0]8C[Twrwr[1:0]]</a> combine to specify a 4-bit value, <a href="#">Twrrwr[3:0]</a> , when <a href="#">F2x[1, 0]94[Ddr3Mode]=1</a> . See <a href="#">F2x[1, 0]8C[Twrwr[1:0]]</a> .														
9:8	<b>Twrrd[3:2]: write to read DIMM termination turnaround.</b> Read-write. This field along with <a href="#">F2x[1, 0]8C[Twrrd[1:0]]</a> combine to specify a 4-bit value, <a href="#">Twrrd[3:0]</a> , when <a href="#">F2x[1, 0]94[Ddr3Mode]=1</a> . See <a href="#">F2x[1, 0]8C[Twrrd[1:0]]</a> .														
7:4	Reserved.														
3:0	<p><b>RdPtrInit: read pointer initial value.</b> Read-write. 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. This field should be written prior to DRAM initialization. See <a href="#">F2x[1, 0]A8[DataTxFifoWrDly]</a>.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Read to Write Pointer Separation</th> </tr> </thead> <tbody> <tr> <td>0000b - 0010b</td> <td>Reserved</td> </tr> <tr> <td>0011b</td> <td>2.5 MEMCLKs (For DDR3, this encoding is reserved.)</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>Revision B: Reserved, Revision C and later: 1 MEMCLK</td> </tr> <tr> <td>0111b - 1111b</td> <td>Reserved</td> </tr> </tbody> </table> <p>For revision B, BIOS should program this to 0101b. For revision C and later, BIOS should program this as follows:  If (Revision C3 &amp;&amp; (<a href="#">CPUID Fn8000_0001_EBX[PkgType]</a>==(0010b    0100b)) &amp;&amp; <a href="#">F3x1F0[NbP-state] !=000b</a>) then 0100b; else 0110b.</p>	Bits	Read to Write Pointer Separation	0000b - 0010b	Reserved	0011b	2.5 MEMCLKs (For DDR3, this encoding is reserved.)	0100b	2 MEMCLKs	0101b	1.5 MEMCLKs	0110b	Revision B: Reserved, Revision C and later: 1 MEMCLK	0111b - 1111b	Reserved
Bits	Read to Write Pointer Separation														
0000b - 0010b	Reserved														
0011b	2.5 MEMCLKs (For DDR3, this encoding is reserved.)														
0100b	2 MEMCLKs														
0101b	1.5 MEMCLKs														
0110b	Revision B: Reserved, Revision C and later: 1 MEMCLK														
0111b - 1111b	Reserved														

### F2x[1, 0]7C DRAM Initialization Register

Reset: 0000 0000h. See 2.8.1 [\[DCT Configuration Registers\]](#) for general programming information about DCT configuration registers.

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. For registered DIMMs, BIOS should follow the recommendations for reset usage in the JEDEC RDIMM specification during the initialization sequence. After completing the sequence, BIOS clears [EnDramInit](#) to complete DRAM initialization. BIOS should not assert [LDTSTOP\\_L](#) while [EnDramInit](#) is set. Note: setting more than one of the command bits in this register ([SendControlWord](#), [SendMrsCmd](#), [SendAutoRefresh](#), and [SendPchgAll](#)) 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 asserts memory reset and deasserts <a href="#">CKE</a> when this bit is set. BIOS must wait until <a href="#">F2x[1, 0]98[DctAccessDone] = 1</a> before programming <a href="#">AssertCke=1</a> and <a href="#">DeassertMemRstX=1</a> . BIOS must clear this bit after DRAM initialization is complete. BIOS must not set this bit on a DCT with no attached DIMMs. See 2.8.9.6.1 <a href="#">[Software DDR2 Device Initialization]</a> and 2.8.9.6.2 <a href="#">[Software DDR3 Device Initialization]</a> .
30	<b>SendControlWord: send control word.</b> Read; write-1-only. 1= The DCT sends a control word to a chip select pair defined in <a href="#">F2x[1, 0]A8[CtrlWordCS]</a> . This bit is cleared by hardware after the command completes. This bit is valid only when <a href="#">F2x[1, 0]94[Ddr3Mode] = 1</a> and <a href="#">F2x[1, 0]90[UnbufDimm] = 0</a> .

29	<b>SendZQCmd: send ZQ command.</b> Read; write-1-only. 1=The DCT sends the ZQ calibration command. Revision C and later: 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 $F2x[1, 0]94[Ddr3Mode]=1$ and $EnDramInit=1$ .										
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/EMRS command.</b> Read; write-1-only. 1=The DCT sends the MRS or EMRS commands defined by the MrsAddress and MrsBank fields of this register. This bit is cleared by hardware after the command completes. This cannot be used for OCD-adjust commands.										
25	<b>SendAutoRefresh: send auto refresh command.</b> Read; write-1-only. 1=The DCT sends an auto refresh command. This bit is cleared by hardware after the command completes.										
24	<b>SendPchgAll: send precharge all command.</b> Read; write-1-only. 1=The DCT sends a precharge-all command. This bit is cleared by hardware after the command completes.										
23	Reserved.										
22:20	<b>MrsChipSel: MRS/EMRS command chip select.</b> Read-write. This field specifies which DRAM chip select is used for MRS/EMRS commands. For DDR2 and DDR3 unbuffered DIMMs, this field is valid only when $EnDramInit = 0$ ; otherwise, MRS/EMRS commands are sent to all chip selects. For DDR3 registered DIMMs, this field specifies the chip select used for MR commands for software initialization only; i.e., when $EnDramInit = 1$ . <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>MRS/EMRS command is sent to CS0</td> </tr> <tr> <td>001b</td> <td>MRS/EMRS command is sent to CS1</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>111b</td> <td>MRS/EMRS command is sent to CS7</td> </tr> </tbody> </table>	Bits	Definition	000b	MRS/EMRS command is sent to CS0	001b	MRS/EMRS command is sent to CS1	...	...	111b	MRS/EMRS command is sent to CS7
Bits	Definition										
000b	MRS/EMRS command is sent to CS0										
001b	MRS/EMRS command is sent to CS1										
...	...										
111b	MRS/EMRS command is sent to CS7										
19	Reserved.										
18:16	<b>MrsBank: bank address for MRS/EMRS commands.</b> Read-write. This field specifies the data driven on the DRAM bank pins for MRS and EMRS commands.										
15:0	<b>MrsAddress: address for MRS/EMRS commands.</b> Read-write. This field specifies the data driven on the DRAM address pins 15-0 for MRS and EMRS commands.										

### F2x[1, 0]80 DRAM Bank Address Mapping Register

Reset: 0000 0000h. See [2.8.1 \[DCT Configuration Registers\]](#) for general programming information about DCT configuration registers.

These fields specify DIMM configuration information. Dimm0AddrMap applies to each physical DIMM of logical DIMM 0 (where logical DIMM numbers are specified by [\[The DRAM CS Base Address Registers\] F2x\[1, 0\]\[5C:40\]](#)), and so forth. These fields are required to be programmed per the following table, based on the DRAM device size and width information of the DIMM. [Table 85](#), for DDR2, [Table 86](#), for DDR3, shows the bit numbers for each position when the DCTs are operating in 64-bit mode (unganged); for 128-bit mode (ganged), address bit 3 delineates between the two channels and the address bit numbers in the table must be incremented by one.

Bits	Description
31:16	Reserved.
15:12	<b>Dimm3AddrMap: DIMM 3 address map.</b> Read-write.
11:8	<b>Dimm2AddrMap: DIMM 2 address map.</b> Read-write.
7:4	<b>Dimm1AddrMap: DIMM 1 address map.</b> Read-write.
3:0	<b>Dimm0AddrMap: DIMM 0 address map.</b> Read-write.

**Table 85: DDR2 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	128 MB	256Mb, x16	x	13	12	Row	x	x	x	17	16	15	14	26	25	24	23	22	21	20	19	18
						Col	x	x	x	x	x	AP	x	11	10	9	8	7	6	5	4	3
0001b	256MB	256Mb, x8 512Mb, x16	x	14	13	Row	x	x	x	17	16	15	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	x	14	13	Row	x	x	17	16	15	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	512MB	256Mb, x4	x	15	14	Row	x	x	x	17	16	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
0100b	512MB	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
0101b	1GB	1G, x8 2G, 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	1GB	512Mb, x4	x	15	14	Row	x	x	17	16	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
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	2GB	1Gb, x4	16	15	14	Row	x	x	17	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
1001b	4GB	2Gb, x4	16	15	14	Row	x	17	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
1010b	4GB	4Gb, x8	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	4Gb, x4	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

**Table 86: 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

**Table 86: 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	
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	1GB	512Mb, x4	16	15	14	Row	x	x	x	17	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
0111b	2GB	2Gb, x8	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	2GB	1Gb, x4	16	15	14	Row	x	x	17	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
1001b	4GB	2Gb, x4	16	15	14	Row	x	17	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
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	4Gb, x4 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

**F2x[1, 0]84 DRAM MRS Register**

Reset: 0000 0004h. All fields of this register are programmed into the DRAM device mode registers, MR[3, 2, 1, 0], for each DDR3 DRAM device during the DRAM initialization process. For DDR2-1066, only the Twr field is applicable.

See 2.8.1 [DCT Configuration Registers] for general programming information about DCT configuration registers.

Bits	Description
31:27	Reserved.
26	<b>MprEn: multi purpose register enable.</b> Read-write. 0=The multi purpose register is disabled. 1=The DDR3-defined multi purpose register enabled; data from subsequent read transactions come from the multi purpose register.
25:24	<b>MprLoc: multi purpose register address location.</b> Read-write. If MprEn=0, this field is ignored. 00b=Load the DDR3-defined pattern into multi purpose register. All other encoding are reserved.



23	<p><b>PchgPDMoSel: precharge power down mode select.</b> Read-write. This specifies how a chip select enters and exits power down mode when enabled by <code>F2x[1, 0]94[PowerDownEn]</code> and <code>F2x[1, 0]94[PowerDownMode]</code>.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td>DDR3-defined deep power down mode; The DCT issues MRS command(s) to the DRAM during initialization to shut off DLLs when in power down. The DCT issues the first valid read, read with auto-precharge, or synchronous ODT command a minimum of 24ns (DDR3-defined tXPDLL) after precharge power down exit.</td> </tr> <tr> <td>1b</td> <td>Dynamic deep/shallow power down mode; The DCT dynamically issues MRS command(s) to the DRAM to specify either deep or shallow powerdown. The DCT specifies shallow power down when one of two chip selects has recently been active. It specifies deep power down when all chip selects have been idle.</td> </tr> </tbody> </table> <p>This bit is valid only when <code>F2x[1, 0]94[Ddr3Mode] = 1</code>.</p>	Bit	Description	0b	DDR3-defined deep power down mode; The DCT issues MRS command(s) to the DRAM during initialization to shut off DLLs when in power down. The DCT issues the first valid read, read with auto-precharge, or synchronous ODT command a minimum of 24ns (DDR3-defined tXPDLL) after precharge power down exit.	1b	Dynamic deep/shallow power down mode; The DCT dynamically issues MRS command(s) to the DRAM to specify either deep or shallow powerdown. The DCT specifies shallow power down when one of two chip selects has recently been active. It specifies deep power down when all chip selects have been idle.				
Bit	Description										
0b	DDR3-defined deep power down mode; The DCT issues MRS command(s) to the DRAM during initialization to shut off DLLs when in power down. The DCT issues the first valid read, read with auto-precharge, or synchronous ODT command a minimum of 24ns (DDR3-defined tXPDLL) after precharge power down exit.										
1b	Dynamic deep/shallow power down mode; The DCT dynamically issues MRS command(s) to the DRAM to specify either deep or shallow powerdown. The DCT specifies shallow power down when one of two chip selects has recently been active. It specifies deep power down when all chip selects have been idle.										
22:20	<p><b>Tcwl: CAS write latency.</b> Read-write. This specifies the number of clock cycles from internal write command to first write data in at the DRAM.</p> <table border="1"> <tbody> <tr> <td>000b</td> <td>5 clocks (MEMCLK &gt;= 2.5ns)</td> </tr> <tr> <td>001b</td> <td>6 clocks (2.5ns &gt; MEMCLK &gt;= 1.875ns)</td> </tr> <tr> <td>010b</td> <td>7 clocks (1.875ns &gt; MEMCLK &gt;= 1.5ns)</td> </tr> <tr> <td>011b</td> <td>8 clocks (1.5ns &gt; MEMCLK &gt;= 1.25ns)</td> </tr> <tr> <td>100b - 111b</td> <td>Reserved.</td> </tr> </tbody> </table> <p>For DDR2, write CAS latency is always read CAS latency minus 1.</p>	000b	5 clocks (MEMCLK >= 2.5ns)	001b	6 clocks (2.5ns > MEMCLK >= 1.875ns)	010b	7 clocks (1.875ns > MEMCLK >= 1.5ns)	011b	8 clocks (1.5ns > MEMCLK >= 1.25ns)	100b - 111b	Reserved.
000b	5 clocks (MEMCLK >= 2.5ns)										
001b	6 clocks (2.5ns > MEMCLK >= 1.875ns)										
010b	7 clocks (1.875ns > MEMCLK >= 1.5ns)										
011b	8 clocks (1.5ns > MEMCLK >= 1.25ns)										
100b - 111b	Reserved.										
19	<p><b>SRT: self refresh temperature range.</b> Read-write. Specifies the SRT range for the DRAM devices. 0=Normal operating temperature range. 1=Extended operating temperature range. If ASR=1 then SRT must be 0.</p>										
18	<p><b>ASR: auto self refresh.</b> Read-write. Specifies the ASR mode for the DRAM devices. 1=DDR3 SDRAM automatically provides self refresh entry and power management functions for all supported operating temperature values. 0=ASR is disabled and SRT is used.</p>										
17:14	Reserved.										
13	<p><b>Qoff: output disable.</b> Read-write. Specifies the QOFF value for the DRAM devices. 0=Output buffers enabled. 1=Output buffers disabled.</p>										
12	Reserved.										
11:10	<p><b>DramTermDyn: DRAM dynamic termination.</b> Read-write. This specifies the programming of the DRAM dynamic termination value for writes when the MRS command is issued to configure MR2 during DDR3 DRAM initialization (<code>F2x[1, 0]90[InitDram]</code>). BIOS must enable dynamic termination when there are 2 DIMMs on a channel. This field is valid only when <code>F2x[1, 0]94[Ddr3Mode] = 1</code>.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>DDR3 definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Dynamic termination for writes disabled</td> </tr> <tr> <td>01b</td> <td>RZQ/4</td> </tr> <tr> <td>10b</td> <td>RZQ/2 (Recommended for 2 DIMMs)</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	DDR3 definition	00b	Dynamic termination for writes disabled	01b	RZQ/4	10b	RZQ/2 (Recommended for 2 DIMMs)	11b	Reserved
Bits	DDR3 definition										
00b	Dynamic termination for writes disabled										
01b	RZQ/4										
10b	RZQ/2 (Recommended for 2 DIMMs)										
11b	Reserved										

9:7	<p><b>DramTerm: DRAM nominal termination.</b> Read-write. This specifies the programming of the DRAM nominal termination value when the MRS command is issued to configure MR1 during DDR3 DRAM initialization (<math>F2x[1, 0]90[InitDram]</math>). This field is valid only when <math>F2x[1, 0]94[Ddr3Mode] = 1</math>. For DDR2 termination values, see <math>F2x[1, 0]90[DramTerm]</math>.</p> <table> <thead> <tr> <th>Bits</th> <th>DDR3 definition</th> <th>Bits</th> <th>DDR3 definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>On die termination disabled</td> <td>100b</td> <td>RZQ/12</td> </tr> <tr> <td>001b</td> <td>RZQ/4</td> <td>101b</td> <td>RZQ/8</td> </tr> <tr> <td>010b</td> <td>RZQ/2</td> <td>110b</td> <td>Reserved</td> </tr> <tr> <td>011b</td> <td>RZQ/6</td> <td>111b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	DDR3 definition	Bits	DDR3 definition	000b	On die termination disabled	100b	RZQ/12	001b	RZQ/4	101b	RZQ/8	010b	RZQ/2	110b	Reserved	011b	RZQ/6	111b	Reserved																
Bits	DDR3 definition	Bits	DDR3 definition																																		
000b	On die termination disabled	100b	RZQ/12																																		
001b	RZQ/4	101b	RZQ/8																																		
010b	RZQ/2	110b	Reserved																																		
011b	RZQ/6	111b	Reserved																																		
6:4	<p><b>Twr: write recovery.</b> Read-write. This specifies the minimum time from the last data write until the chip-select bank precharge; this is the WR field in the DDR3 specification. See <math>F2x[1, 0]88[Twr]</math> for DDR2 values. This field specifies the encodings for DDR2-1066 when <math>F2x[1, 0]94[Ddr3Mode] = 0</math> and when <math>F2x[1, 0]94[MemClkFreq]=100b</math>.</p> <table> <thead> <tr> <th>Bits</th> <th>DDR2-1066 definition</th> <th>Bits</th> <th>DDR3 definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Reserved</td> <td>000b</td> <td>Reserved</td> </tr> <tr> <td>001b</td> <td>5 MEMCLK cycles</td> <td>001b</td> <td>5 MEMCLK cycles</td> </tr> <tr> <td>010b</td> <td>6 MEMCLK cycles</td> <td>010b</td> <td>6 MEMCLK cycles</td> </tr> <tr> <td>011b</td> <td>Reserved</td> <td>011b</td> <td>7 MEMCLK cycles</td> </tr> <tr> <td>100b</td> <td>8 MEMCLK cycles</td> <td>100b</td> <td>8 MEMCLK cycles</td> </tr> <tr> <td>101b</td> <td>Reserved</td> <td>101b</td> <td>10 MEMCLK cycles</td> </tr> <tr> <td>110b</td> <td>Reserved</td> <td>110b</td> <td>12 MEMCLK cycles</td> </tr> <tr> <td>111b</td> <td>Reserved</td> <td>111b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	DDR2-1066 definition	Bits	DDR3 definition	000b	Reserved	000b	Reserved	001b	5 MEMCLK cycles	001b	5 MEMCLK cycles	010b	6 MEMCLK cycles	010b	6 MEMCLK cycles	011b	Reserved	011b	7 MEMCLK cycles	100b	8 MEMCLK cycles	100b	8 MEMCLK cycles	101b	Reserved	101b	10 MEMCLK cycles	110b	Reserved	110b	12 MEMCLK cycles	111b	Reserved	111b	Reserved
Bits	DDR2-1066 definition	Bits	DDR3 definition																																		
000b	Reserved	000b	Reserved																																		
001b	5 MEMCLK cycles	001b	5 MEMCLK cycles																																		
010b	6 MEMCLK cycles	010b	6 MEMCLK cycles																																		
011b	Reserved	011b	7 MEMCLK cycles																																		
100b	8 MEMCLK cycles	100b	8 MEMCLK cycles																																		
101b	Reserved	101b	10 MEMCLK cycles																																		
110b	Reserved	110b	12 MEMCLK cycles																																		
111b	Reserved	111b	Reserved																																		
3:2	<p><b>DrvImpCtrl: drive impedance control.</b> Read-write. This field specifies impedance of the DRAM output driver. This field is valid only when <math>F2x[1, 0]94[Ddr3Mode] = 1</math>.</p> <table> <thead> <tr> <th>Bits</th> <th>DDR3 definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>40 ohm driver; <math>R_{on40} = R_{zq}/6</math> (40 ohm with nominal <math>R_{zq}=240</math> ohms)</td> </tr> <tr> <td>01b</td> <td>34 ohm driver; <math>R_{on34} = R_{zq}/7</math> (34 ohm with nominal <math>R_{zq}=240</math> ohms)</td> </tr> <tr> <td>10b</td> <td>Reserved for 30 ohm driver; <math>R_{on30} = R_{zq}/8</math> (30 ohm with nominal <math>R_{zq}=240</math> ohms)</td> </tr> <tr> <td>11b</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	DDR3 definition	00b	40 ohm driver; $R_{on40} = R_{zq}/6$ (40 ohm with nominal $R_{zq}=240$ ohms)	01b	34 ohm driver; $R_{on34} = R_{zq}/7$ (34 ohm with nominal $R_{zq}=240$ ohms)	10b	Reserved for 30 ohm driver; $R_{on30} = R_{zq}/8$ (30 ohm with nominal $R_{zq}=240$ ohms)	11b	Reserved																										
Bits	DDR3 definition																																				
00b	40 ohm driver; $R_{on40} = R_{zq}/6$ (40 ohm with nominal $R_{zq}=240$ ohms)																																				
01b	34 ohm driver; $R_{on34} = R_{zq}/7$ (34 ohm with nominal $R_{zq}=240$ ohms)																																				
10b	Reserved for 30 ohm driver; $R_{on30} = R_{zq}/8$ (30 ohm with nominal $R_{zq}=240$ ohms)																																				
11b	Reserved																																				
1:0	<p><b>BurstCtrl: burst length control.</b> Read-write. Specifies the number of sequential beats of DQ related to one read or write command. This field interacts with <math>F2x[1, 0]90[Width128]</math> as follows:</p> <ul style="list-style-type: none"> <li>• If <math>Width128 = 0</math>, then BIOS should program this field to 00b (8-beat burst length; 64-byte access).</li> <li>• If <math>Width128 = 1</math>, then BIOS should program this field to 10b (4-beat burst length; 64-byte access).</li> </ul> <p>All other encodings are reserved.</p>																																				

### F2x[1, 0]88 DRAM Timing Low Register

Reset: FF00 0000h. See 2.8.1 [DCT Configuration Registers] for general programming information about DCT configuration registers.

Bits	Description																												
31:24	<p><b>MemClkDis: MEMCLK disable.</b> Read-write. 1=Tristate the specified MEMCLK driver pads for power savings. The bits of this field map to pads as follows:</p> <table border="0"> <thead> <tr> <th><u>Bit</u></th> <th><u>Pad</u></th> <th><u>Bit</u></th> <th><u>Pad</u></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>MEMCLK[1,0]_H[0]</td> <td>4</td> <td>MEMCLK[1,0]_H[4]</td> </tr> <tr> <td>1</td> <td>MEMCLK[1,0]_H[1]</td> <td>5</td> <td>MEMCLK[1,0]_H[5]</td> </tr> <tr> <td>2</td> <td>MEMCLK[1,0]_H[2]</td> <td>6</td> <td>MEMCLK[1,0]_H[6]</td> </tr> <tr> <td>3</td> <td>MEMCLK[1,0]_H[3]</td> <td>7</td> <td>MEMCLK[1,0]_H[7]</td> </tr> </tbody> </table> <p>See section 2.8.2 [DDR Pad to Processor Pin Mapping].</p>	<u>Bit</u>	<u>Pad</u>	<u>Bit</u>	<u>Pad</u>	0	MEMCLK[1,0]_H[0]	4	MEMCLK[1,0]_H[4]	1	MEMCLK[1,0]_H[1]	5	MEMCLK[1,0]_H[5]	2	MEMCLK[1,0]_H[2]	6	MEMCLK[1,0]_H[6]	3	MEMCLK[1,0]_H[3]	7	MEMCLK[1,0]_H[7]								
<u>Bit</u>	<u>Pad</u>	<u>Bit</u>	<u>Pad</u>																										
0	MEMCLK[1,0]_H[0]	4	MEMCLK[1,0]_H[4]																										
1	MEMCLK[1,0]_H[1]	5	MEMCLK[1,0]_H[5]																										
2	MEMCLK[1,0]_H[2]	6	MEMCLK[1,0]_H[6]																										
3	MEMCLK[1,0]_H[3]	7	MEMCLK[1,0]_H[7]																										
23:22	<p><b>Trrd: row to row delay (or RAS to RAS delay).</b> Read-write. This specifies the minimum time between activate commands to different chip-select banks. The definition of this field varies with the DDR type (F2x[1, 0]94[Ddr3Mode]) and when F2x[1, 0]94[MemClkFreq]=100b to support DDR2-1066.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>DDR2 definition</u></th> <th><u>Bits</u></th> <th><u>DDR2-1066/DDR3 definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>2 clocks</td> <td>00b</td> <td>4 clocks</td> </tr> <tr> <td>01b</td> <td>3 clocks</td> <td>01b</td> <td>5 clocks</td> </tr> <tr> <td>10b</td> <td>4 clocks</td> <td>10b</td> <td>6 clocks</td> </tr> <tr> <td>11b</td> <td>5 clocks</td> <td>11b</td> <td>7 clocks</td> </tr> </tbody> </table>	<u>Bits</u>	<u>DDR2 definition</u>	<u>Bits</u>	<u>DDR2-1066/DDR3 definition</u>	00b	2 clocks	00b	4 clocks	01b	3 clocks	01b	5 clocks	10b	4 clocks	10b	6 clocks	11b	5 clocks	11b	7 clocks								
<u>Bits</u>	<u>DDR2 definition</u>	<u>Bits</u>	<u>DDR2-1066/DDR3 definition</u>																										
00b	2 clocks	00b	4 clocks																										
01b	3 clocks	01b	5 clocks																										
10b	4 clocks	10b	6 clocks																										
11b	5 clocks	11b	7 clocks																										
21:20	<p><b>Twr: write recovery time.</b> Read-write. This specifies the minimum time from the last data write until the chip-select bank precharge. This is only valid if F2x[1, 0]94[Ddr3Mode]=0 and when F2x[1, 0]94[MemClkFreq]≠100b; otherwise, bit 21 is reserved and bit 20 becomes part of the Trc field below. For DDR2-1066 and DDR3 see F2x[1, 0]84[Twr].</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>DDR2 definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>3 clocks</td> </tr> <tr> <td>01b</td> <td>4 clocks</td> </tr> <tr> <td>10b</td> <td>5 clocks</td> </tr> <tr> <td>11b</td> <td>6 clocks</td> </tr> </tbody> </table>	<u>Bits</u>	<u>DDR2 definition</u>	00b	3 clocks	01b	4 clocks	10b	5 clocks	11b	6 clocks																		
<u>Bits</u>	<u>DDR2 definition</u>																												
00b	3 clocks																												
01b	4 clocks																												
10b	5 clocks																												
11b	6 clocks																												
19:16 DDR2 20:16 DDR2 1066/ DDR3	<p><b>Trc: row cycle time.</b> Read-write. This specifies the minimum time from and activate command to another activate command or an auto-refresh command, all to the same chip-select bank. This size and definition of this field varies with the DDR type (F2x[1, 0]94[Ddr3Mode]) and when F2x[1, 0]94[MemClkFreq]=100b to support DDR2-1066.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>DDR2 definition</u></th> <th><u>Bits</u></th> <th><u>DDR2-1066/DDR3 definition</u></th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>11 clocks</td> <td>00h</td> <td>11 clocks</td> </tr> <tr> <td>1h</td> <td>12 clocks</td> <td>01h</td> <td>12 clocks</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>Fh</td> <td>26 clocks</td> <td>1Fh</td> <td>42 clocks</td> </tr> </tbody> </table>	<u>Bits</u>	<u>DDR2 definition</u>	<u>Bits</u>	<u>DDR2-1066/DDR3 definition</u>	0h	11 clocks	00h	11 clocks	1h	12 clocks	01h	12 clocks	...	...	...	...	Fh	26 clocks	1Fh	42 clocks								
<u>Bits</u>	<u>DDR2 definition</u>	<u>Bits</u>	<u>DDR2-1066/DDR3 definition</u>																										
0h	11 clocks	00h	11 clocks																										
1h	12 clocks	01h	12 clocks																										
...	...	...	...																										
Fh	26 clocks	1Fh	42 clocks																										
15:12	<p><b>Tras: row active strobe.</b> Read-write. This specifies the minimum time from an activate command to a precharge command, both to the same chip-select bank. The definition of this field varies with the DDR type (F2x[1, 0]94[Ddr3Mode]) and when F2x[1, 0]94[MemClkFreq]=100b to support DDR2-1066.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>DDR2 definition</u></th> <th><u>Bits</u></th> <th><u>DDR2-1066/DDR3 definition</u></th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>Reserved</td> <td>0h</td> <td>15 clocks</td> </tr> <tr> <td>1h</td> <td>Reserved</td> <td>1h</td> <td>16 clocks</td> </tr> <tr> <td>2h</td> <td>5 clocks</td> <td>2h</td> <td>17 clocks</td> </tr> <tr> <td>3h</td> <td>6 clocks</td> <td>3h</td> <td>18 clocks</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>Fh</td> <td>18 clocks</td> <td>Fh</td> <td>30 clocks</td> </tr> </tbody> </table>	<u>Bits</u>	<u>DDR2 definition</u>	<u>Bits</u>	<u>DDR2-1066/DDR3 definition</u>	0h	Reserved	0h	15 clocks	1h	Reserved	1h	16 clocks	2h	5 clocks	2h	17 clocks	3h	6 clocks	3h	18 clocks	...	...	...	...	Fh	18 clocks	Fh	30 clocks
<u>Bits</u>	<u>DDR2 definition</u>	<u>Bits</u>	<u>DDR2-1066/DDR3 definition</u>																										
0h	Reserved	0h	15 clocks																										
1h	Reserved	1h	16 clocks																										
2h	5 clocks	2h	17 clocks																										
3h	6 clocks	3h	18 clocks																										
...	...	...	...																										
Fh	18 clocks	Fh	30 clocks																										

11:10	<p><b>Trtp: read to precharge time.</b> Read-write. Read CAS to Precharge. This specifies the earliest time a page can be closed after having been read. Satisfying this parameter ensures read data is not lost due to a premature precharge. The size and definition of this field varies with the DDR type, <math>F2x[1, 0]94[Ddr3Mode]</math>. The recommended value for this field varies with DDR type and speed. This field should not be confused with tRTP, which is the internal DRAM timing as is specified by the DRAM data sheet and also SPD byte 38.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>DDR2 definition</u></th> <th><u>Recommended value.</u></th> </tr> </thead> <tbody> <tr> <td>0xb</td> <td>2 clocks for burst length of 32 bytes</td> <td>(DDR400, DDR533)</td> </tr> <tr> <td></td> <td>4 clocks for burst length of 64 bytes</td> <td>(DDR400, DDR533)</td> </tr> <tr> <td>1xb</td> <td>3 clocks for burst length of 32 bytes</td> <td>(DDR667, DDR800, Revision B: DDR1066)</td> </tr> <tr> <td></td> <td>5 clocks for burst length of 64 bytes</td> <td>(DDR667, DDR800, Revision B: DDR1066)</td> </tr> <tr> <td colspan="3"> </td> </tr> <tr> <td colspan="3" style="text-align: center;"><u>DDR2-1066 for revision C and later and</u></td> </tr> <tr> <th><u>Bits</u></th> <th><u>DDR3 definition</u></th> <th><u>Recommended value.</u></th> </tr> <tr> <td>00b</td> <td>4 clocks</td> <td>(DDR800, DDR1066) See note.</td> </tr> <tr> <td>01b</td> <td>5 clocks</td> <td>(DDR1333)</td> </tr> <tr> <td>10b</td> <td>6 clocks</td> <td></td> </tr> <tr> <td>11b</td> <td>7 clocks</td> <td></td> </tr> </tbody> </table> <p>Note: For revision C and later revisions, the recommended DDR2-1066 values for Trtp are provided for a burst length of 32 bytes. For a burst length of 64 bytes, increase the listed value by 2 clocks.</p>	<u>Bits</u>	<u>DDR2 definition</u>	<u>Recommended value.</u>	0xb	2 clocks for burst length of 32 bytes	(DDR400, DDR533)		4 clocks for burst length of 64 bytes	(DDR400, DDR533)	1xb	3 clocks for burst length of 32 bytes	(DDR667, DDR800, Revision B: DDR1066)		5 clocks for burst length of 64 bytes	(DDR667, DDR800, Revision B: DDR1066)	 			<u>DDR2-1066 for revision C and later and</u>			<u>Bits</u>	<u>DDR3 definition</u>	<u>Recommended value.</u>	00b	4 clocks	(DDR800, DDR1066) See note.	01b	5 clocks	(DDR1333)	10b	6 clocks		11b	7 clocks	
<u>Bits</u>	<u>DDR2 definition</u>	<u>Recommended value.</u>																																			
0xb	2 clocks for burst length of 32 bytes	(DDR400, DDR533)																																			
	4 clocks for burst length of 64 bytes	(DDR400, DDR533)																																			
1xb	3 clocks for burst length of 32 bytes	(DDR667, DDR800, Revision B: DDR1066)																																			
	5 clocks for burst length of 64 bytes	(DDR667, DDR800, Revision B: DDR1066)																																			
<u>DDR2-1066 for revision C and later and</u>																																					
<u>Bits</u>	<u>DDR3 definition</u>	<u>Recommended value.</u>																																			
00b	4 clocks	(DDR800, DDR1066) See note.																																			
01b	5 clocks	(DDR1333)																																			
10b	6 clocks																																				
11b	7 clocks																																				
9:7	<p><b>Trp: row precharge time.</b> Read-write. This specifies the minimum time from a precharge command to an activate command or auto-refresh command, both to the same bank. This size and definition of this field varies with the DDR type (<math>F2x[1, 0]94[Ddr3Mode]</math>) and when <math>F2x[1, 0]94[MemClk-Freq]=100b</math> to support DDR2-1066.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>DDR2 definition</u></th> <th><u>Bits</u></th> <th><u>DDR2-1066/DDR3 definition</u></th> </tr> </thead> <tbody> <tr> <td>00xb</td> <td>3 clocks</td> <td>000b</td> <td>5 clocks</td> </tr> <tr> <td>01xb</td> <td>4 clocks</td> <td>001b</td> <td>6 clocks</td> </tr> <tr> <td>10xb</td> <td>5 clocks</td> <td>...</td> <td>...</td> </tr> <tr> <td>11xb</td> <td>6 clocks</td> <td>110b</td> <td>11 clocks</td> </tr> <tr> <td></td> <td></td> <td>111b</td> <td>12 clocks</td> </tr> </tbody> </table>	<u>Bits</u>	<u>DDR2 definition</u>	<u>Bits</u>	<u>DDR2-1066/DDR3 definition</u>	00xb	3 clocks	000b	5 clocks	01xb	4 clocks	001b	6 clocks	10xb	5 clocks	...	...	11xb	6 clocks	110b	11 clocks			111b	12 clocks												
<u>Bits</u>	<u>DDR2 definition</u>	<u>Bits</u>	<u>DDR2-1066/DDR3 definition</u>																																		
00xb	3 clocks	000b	5 clocks																																		
01xb	4 clocks	001b	6 clocks																																		
10xb	5 clocks	...	...																																		
11xb	6 clocks	110b	11 clocks																																		
		111b	12 clocks																																		

6:4	<b>Trcd: RAS to CAS delay.</b> Read-write. This specifies the time from an activate command to a read/write command, both to the same bank. This size and definition of this field varies with the DDR type ( $F2x[1, 0]94[Ddr3Mode]$ ) and when $F2x[1, 0]94[MemClkFreq]=100b$ to support DDR2-1066.			
	<u>Bits</u>	<u>DDR2 definition</u>	<u>Bits</u>	<u>DDR2-1066/DDR3 definition</u>
	x00b	3 clocks	000b	5 clocks
	x01b	4 clocks	001b	6 clocks
	x10b	5 clocks	...	...
	x11b	6 clocks	110b	11 clocks
			111b	12 clocks
3:0	<b>Tcl: CAS latency.</b> Read-write. This specifies the time from the CAS assertion for a read cycle until data return (from the perspective of the DRAM devices). The DCT adjusts these latencies appropriately for registered DIMMs. The definition of this field varies with the DDR type specified by $F2x[1, 0]94[Ddr3Mode]$ .			
	<u>Bits</u>	<u>DDR2 definition</u>	<u>Bits</u>	<u>DDR3 definition</u>
	0000b	Reserved	0000b	4 clocks
	0001b	Reserved	0001b	5 clocks
	0010b	3 clocks	0010b	6 clocks
	0011b	4 clocks	...	...
	0100b	5 clocks	0111b	11 clocks
	0101b	6 clocks	1000b	12 clocks
	0110b	7 clocks	1001b - 1111b	Reserved
	0111b - 1111b	Reserved		

### F2x[1, 0]8C DRAM Timing High Register

Reset: 0000 0000h. See 2.8.1 [DCT Configuration Registers] for general programming information about DCT configuration registers.

Bits	Description
31:29	<b>Trfc3: auto-refresh row cycle time for logical DIMM 3.</b> Read-write. See Trfc0.
28:26	<b>Trfc2: auto-refresh row cycle time for logical DIMM 2.</b> Read-write. See Trfc0.
25:23	<b>Trfc1: auto-refresh row cycle time for logical DIMM 1.</b> Read-write. See Trfc0.
22:20	<b>Trfc0: auto-refresh row cycle time for logical DIMM 0.</b> Read-write. This specifies the minimum time from a refresh command to the next valid command, except NOP or DES. DIMM numbers are specified by [The DRAM CS Base Address Registers] $F2x[1, 0][5C:40]$ and map to chip select pairs. The recommended programming of this register varies based on DRAM density and speed. The definition of this field varies with the DDR type, $F2x[1, 0]94[Ddr3Mode]$ .
	<u>Bits</u> <u>DDR2 definition</u> <u>DDR3 definition</u>
	000b                      75 ns (all speeds, 256 Mbit)                      Reserved
	001b                      105 ns (all speeds, 512 Mbit)                      90 ns (all speeds, 512 Mbit)
	010b                      127.5 ns (all speeds, 1 Gbit)                      110 ns (all speeds, 1 Gbit)
	011b                      195 ns (all speeds, 2 Gbit)                      160 ns (all speeds, 2 Gbit)
	100b                      327.5 ns (all speeds, 4 Gbit)                      300 ns (all speeds, 4 Gbit)
	101b                      Reserved                      350 ns (all speeds, 8 Gbit)
	110b-111b                      Reserved                      Reserved
19	Reserved.
18	<b>DisAutoRefresh: disable automatic refresh.</b> Read-write. 1=Automatic refresh is disabled. See 2.8.9.9.1 [Write Levelization Training] and 2.8.9.9.6 [Continuous Pattern Generation].

17:16	<p><b>Tref: refresh rate.</b> Read-write. This specifies the average time between refresh requests to all DRAM devices.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></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 microseconds</td> </tr> <tr> <td>11b</td> <td>Every 3.9 microseconds</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	Undefined behavior.	01b	Reserved	10b	Every 7.8 microseconds	11b	Every 3.9 microseconds														
<u>Bits</u>	<u>Definition</u>																								
00b	Undefined behavior.																								
01b	Reserved																								
10b	Every 7.8 microseconds																								
11b	Every 3.9 microseconds																								
15:14	<p><b>Trdrd[1:0]: read to read timing.</b> Read-write. Trdrd specifies the minimum number of cycles from the last clock of virtual CAS of a first read-burst operation to the clock in which CAS is asserted for a following read-burst operation that is to a different chip select than the first read-burst operation. This field applies to all reads to different chip selects. If consecutive reads involve an ODT change, time must be inserted between the reads to account for (1) turn-around timing and (2) termination timing. This field along with <a href="#">F2x[1, 0]78[Trdrd[3:2]]</a> combine to specify a 4-bit value, Trdrd[3:0], when <a href="#">F2x[1, 0]94[Ddr3Mode]=1</a>. See <a href="#">[The Trdrd (Read to Read Timing)] 2.8.9.5.1</a> for information on how to program this field.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>DDR2 definition</u></th> <th><u>Bits</u></th> <th><u>DDR3 definition</u></th> </tr> </thead> <tbody> <tr> <td>0xb</td> <td>3 clocks</td> <td>000xb</td> <td>3 clocks</td> </tr> <tr> <td>10b</td> <td>4 clocks</td> <td>...</td> <td>...</td> </tr> <tr> <td>11b</td> <td>5 clocks</td> <td>1000b</td> <td>10 clocks</td> </tr> <tr> <td></td> <td></td> <td>1001b - 1111b</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>DDR2 definition</u>	<u>Bits</u>	<u>DDR3 definition</u>	0xb	3 clocks	000xb	3 clocks	10b	4 clocks	...	...	11b	5 clocks	1000b	10 clocks			1001b - 1111b	Reserved				
<u>Bits</u>	<u>DDR2 definition</u>	<u>Bits</u>	<u>DDR3 definition</u>																						
0xb	3 clocks	000xb	3 clocks																						
10b	4 clocks	...	...																						
11b	5 clocks	1000b	10 clocks																						
		1001b - 1111b	Reserved																						
13:12	<p><b>Twrrw[1:0]: write to write timing.</b> Read-write. Twrrw specifies the minimum number of cycles from the last clock of virtual CAS of the first write-burst operation to the clock in which CAS is asserted for a following write-burst operation that changes the enabled terminator. This field applies to all writes to different chip selects. If consecutive writes involve an ODT change, then time must be inserted between them to account for termination timing on DDR devices. This field along with <a href="#">F2x[1, 0]78[Twrwr[3:2]]</a> combine to specify a 4-bit value, Twrrw[3:0], when <a href="#">F2x[1, 0]94[Ddr3Mode]=1</a>.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>DDR2 definition</u></th> <th><u>Bits</u></th> <th><u>DDR3 definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>1 clock</td> <td>0000b - 0001b</td> <td>Reserved</td> </tr> <tr> <td>01b</td> <td>2 clocks</td> <td>0010b</td> <td>2 clocks</td> </tr> <tr> <td>10b</td> <td>3 clocks</td> <td>...</td> <td>...</td> </tr> <tr> <td>11b</td> <td>4 clocks</td> <td>1010b</td> <td>10 clocks</td> </tr> <tr> <td></td> <td></td> <td>1011b - 1111b</td> <td>Reserved</td> </tr> </tbody> </table> <p>See <a href="#">[The Twrrw (Write to Write Timing)] 2.8.9.5.2</a> for information on how to program this field.</p>	<u>Bits</u>	<u>DDR2 definition</u>	<u>Bits</u>	<u>DDR3 definition</u>	00b	1 clock	0000b - 0001b	Reserved	01b	2 clocks	0010b	2 clocks	10b	3 clocks	...	...	11b	4 clocks	1010b	10 clocks			1011b - 1111b	Reserved
<u>Bits</u>	<u>DDR2 definition</u>	<u>Bits</u>	<u>DDR3 definition</u>																						
00b	1 clock	0000b - 0001b	Reserved																						
01b	2 clocks	0010b	2 clocks																						
10b	3 clocks	...	...																						
11b	4 clocks	1010b	10 clocks																						
		1011b - 1111b	Reserved																						

11:10	<p><b>Twrrd[1:0]: write to read DIMM termination turnaround.</b> Read-write. This specifies the minimum number of cycles from the last clock of virtual CAS of the first write operation to the clock in which CAS is asserted for a following read operation involving a memory ODT change on a channel with multiple DIMMs.<sup>1</sup> Time may need to be inserted between these operations to avoid the possibility that there is an overlap of the on die termination timing of the DIMMs.<sup>2</sup> This field along with <a href="#">F2x[1, 0]78[<i>Twrrd</i>[3:2]]</a> combine to specify a 4-bit value, <i>Twrrd</i>[3:0], when <a href="#">F2x[1, 0]94[<i>Ddr3Mode</i>]=1</a>.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>DDR2 definition</u></th> <th><u>Bits</u></th> <th><u>DDR3 definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>1 clock</td> <td>0000b</td> <td>Reserved</td> </tr> <tr> <td>01b</td> <td>2 clocks</td> <td>0001b</td> <td>Reserved</td> </tr> <tr> <td>10b</td> <td>3 clocks</td> <td>0010b</td> <td>2 clocks</td> </tr> <tr> <td>11b</td> <td>4 clocks</td> <td>...</td> <td>...</td> </tr> <tr> <td></td> <td></td> <td>1010b</td> <td>10 clocks</td> </tr> <tr> <td></td> <td></td> <td>1011b - 1111b</td> <td>Reserved</td> </tr> </tbody> </table> <p>Notes:</p> <p>1. For multiple DDR3 DIMMs on a channel, all write operations that are followed by a read require an ODT change and thus the DCT always applies <i>Twrrd</i>.</p> <p>See <a href="#">[The <i>Twrrd</i> (Write to Read DIMM Termination Turn-around)] 2.8.9.5.3</a> for information on how to program this field.</p>	<u>Bits</u>	<u>DDR2 definition</u>	<u>Bits</u>	<u>DDR3 definition</u>	00b	1 clock	0000b	Reserved	01b	2 clocks	0001b	Reserved	10b	3 clocks	0010b	2 clocks	11b	4 clocks	...	...			1010b	10 clocks			1011b - 1111b	Reserved
<u>Bits</u>	<u>DDR2 definition</u>	<u>Bits</u>	<u>DDR3 definition</u>																										
00b	1 clock	0000b	Reserved																										
01b	2 clocks	0001b	Reserved																										
10b	3 clocks	0010b	2 clocks																										
11b	4 clocks	...	...																										
		1010b	10 clocks																										
		1011b - 1111b	Reserved																										
9:8	<p><b>Twtr: internal DRAM write to read command delay.</b> Read-write. This specifies the minimum number of 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. The definition of this field varies with the DDR type, <a href="#">F2x[1, 0]94[<i>Ddr3Mode</i>]</a> and when <a href="#">F2x[1, 0]94[<i>MemClkFreq</i>]=100b</a> to support DDR2-1066.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>DDR2 definition</u></th> <th><u>Bits</u></th> <th><u>DDR2-1066 definition/DDR3 definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Reserved</td> <td>00b</td> <td>4 clocks</td> </tr> <tr> <td>01b</td> <td>1 clocks</td> <td>01b</td> <td>5 clocks</td> </tr> <tr> <td>10b</td> <td>2 clocks</td> <td>10b</td> <td>6 clocks</td> </tr> <tr> <td>11b</td> <td>3 clocks</td> <td>11b</td> <td>7 clocks</td> </tr> </tbody> </table>	<u>Bits</u>	<u>DDR2 definition</u>	<u>Bits</u>	<u>DDR2-1066 definition/DDR3 definition</u>	00b	Reserved	00b	4 clocks	01b	1 clocks	01b	5 clocks	10b	2 clocks	10b	6 clocks	11b	3 clocks	11b	7 clocks								
<u>Bits</u>	<u>DDR2 definition</u>	<u>Bits</u>	<u>DDR2-1066 definition/DDR3 definition</u>																										
00b	Reserved	00b	4 clocks																										
01b	1 clocks	01b	5 clocks																										
10b	2 clocks	10b	6 clocks																										
11b	3 clocks	11b	7 clocks																										
7:4	<p><b>TrwtTO: read to write turnaround for data, DQS contention.</b> Read-write. This specifies the minimum number of cycles from the last clock of virtual CAS of a first read operation to the clock in which CAS is asserted for a following write operation. Time may need to be inserted to ensure there is no bus contention on bidirectional pins.</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>0000b</td> <td>Reserved</td> </tr> <tr> <td>0001b</td> <td>3 clocks</td> </tr> <tr> <td>0010b</td> <td>4 clocks</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>1111b</td> <td>17 clocks</td> </tr> </tbody> </table> <p>See <a href="#">[The <i>TrwtTO</i> (Read-to-Write Turnaround for Data, DQS Contention)] 2.8.9.5.4</a> for information on how to program this field.</p>	<u>Bits</u>	<u>Definition</u>	0000b	Reserved	0001b	3 clocks	0010b	4 clocks	...	...	1111b	17 clocks																
<u>Bits</u>	<u>Definition</u>																												
0000b	Reserved																												
0001b	3 clocks																												
0010b	4 clocks																												
...	...																												
1111b	17 clocks																												



3:0	<b>TrwtWB: read to write turnaround for opportunistic write bursting.</b> Read-write. This specifies the minimum number of cycles from the last virtual CAS of a first read operation to the CAS of a following write operation. The purpose of this field is to hold off write operations until several cycles have elapsed without a read cycle; this may result in performance benefits.			
	<u>Bits</u>	<u>DDR2 definition</u>	<u>Bits</u>	<u>DDR3 definition</u>
	0000b	3 clocks	0000b	3 clocks
	0001b	4 clocks	0001b	4 clocks
	...	...	...	...
	0111b	10 clocks	1111b	18 clocks
	1000b - 1111b	Reserved		
	See <a href="#">[The TrwtWB (Read-to-Write Turnaround for Opportunistic Write Bursting)] 2.8.9.5.5</a> for information on how to program this field.			

### F2x[1, 0]90 DRAM Configuration Low Register

Reset: 0000 0000h, except bit 16 and 27 (see below). See [2.8.1 \[DCT Configuration Registers\]](#) for general programming information about DCT configuration registers.

Bits	Description
31:28	Reserved.
27	<b>DisDllShutdownSR: disable DLL shutdown in self-refresh mode.</b> Revision B, RB-C, BL-C: Reserved. Revision DA-C and C3 and later: Read-write. Reset: 1. 1=Disable the power saving feature of shutting down DDR phy DLLs and tristating CLK pads during DRAM self refresh. 0=Shutdown DLLs during DRAM self refresh. For revision (DA-C    C3    D1 and later G34    D1 and later C32    E), BIOS should program this to 0 after DRAM initialization is complete and after the frequency has been changed when resuming from S3; otherwise BIOS should leave this bit at the default value. See <a href="#">2.8.9.8 [DRAM Channel Frequency Change]</a> .
26:24	Reserved.
23	<b>ForceAutoPchg: force auto precharging.</b> Read-write. 1=Force auto-precharge cycles with every read or write command. BIOS should program this to 0.
22:21	<b>IdleCycLowLimit: idle cycle low limit.</b> Read-write. This 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="#">F2x[1, 0]90[DynPageCloseEn]</a> = 0. <u>Bits</u> <u>Definition</u> 00b    16 clocks 01b    32 clocks 10b    64 clocks 11b    96 clocks
20	<b>DynPageCloseEn: dynamic page close enable.</b> Read-write. 1=The DRAM controller dynamically determines when to close open pages based on the history of that particular page and <a href="#">F2x[1, 0]90[IdleCycLowLimit]</a> . 0=Any open pages not auto-precharged by the DRAM controller are automatically closed after 128 clocks of inactivity.
19	<b>DimmEccEn: DIMM ECC enable.</b> Read-write. 1=ECC checking is capable of being enabled for all DIMMs on the DRAM controller (through <a href="#">F3x44[DrumEccEn]</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. 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. BIOS should not set this bit.

17	<b>EnterSelfRef: enter self refresh command.</b> Read, write-1-only. 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. See 2.8.9.9 [DRAM Training] for information on how to use this bit. See MSRC001_001F[GfxNbPstateDis] for additional requirements when using this bit.															
16	<b>UnbuffDimm: unbuffered DIMM.</b> Read-write or read-only, depending on the product. Reset: value varies based on product. 1=The DRAM controller is connected to unbuffered DIMMs. 0=The DRAM controller is connected to registered DIMMs.															
15:12	<b>X4Dimm: x4 (by 4) DIMMs.</b> Read-write. Each of these bits specifies whether the corresponding logical DIMM (as defined by [The DRAM CS Base Address Registers] F2x[1, 0][5C:40]) is a x4 DIMM or not. The DRAM controller requires this information to make decisions about DIMM signaling. Bit[12] corresponds to logical DIMM 0, bit[13] corresponds to logical DIMM 1, etc. 1=x4 DIMM present. 0=x4 DIMM not present.															
11	<b>Width128: width of DRAM interface in 128-bit mode.</b> Read-write. 1=The DRAM controller interface is 2 DIMMs wide. 0=The DRAM controller interface is 1 DIMM wide.															
10	<p><b>BurstLength32: DRAM burst length set for 32 bytes.</b> Read-write. This specifies the burst length of DRAM accesses and, as a result, the number of data bytes exchanged in each access. 1=32-byte mode. 0=64-byte mode. 32-byte mode may be preferred in platforms that include graphics controllers that generate a lot of 32-byte system memory accesses. 32-byte mode is not supported when the DRAM interface is 128 bits wide; so this bit interacts with F2x[1, 0]90[Width128] as follows:</p> <table border="1"> <thead> <tr> <th>BurstLength32</th> <th>Width128</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>8-beat burst length; 64-byte accesses</td> </tr> <tr> <td>0</td> <td>1</td> <td>4-beat burst length; 64-byte accesses</td> </tr> <tr> <td>1</td> <td>0</td> <td>4-beat burst length; 32-byte accesses</td> </tr> <tr> <td>1</td> <td>1</td> <td>Illegal</td> </tr> </tbody> </table> <p>BurstLength32 is undefined when F2x[1, 0]94[Ddr3Mode]=1.</p>	BurstLength32	Width128	Description	0	0	8-beat burst length; 64-byte accesses	0	1	4-beat burst length; 64-byte accesses	1	0	4-beat burst length; 32-byte accesses	1	1	Illegal
BurstLength32	Width128	Description														
0	0	8-beat burst length; 64-byte accesses														
0	1	4-beat burst length; 64-byte accesses														
1	0	4-beat burst length; 32-byte accesses														
1	1	Illegal														
9	<b>SelfRefRateEn: faster self refresh rate enable.</b> Read-write. 1=Enables high temperature (two times normal) self refresh rate. This bit is reflected in the EMRS(2) command to the DRAM devices. This bit is undefined when F2x[1, 0]94[Ddr3Mode]=1.															
8	<b>ParEn: parity enable.</b> Read-write. 1=Enables address parity computation output, PAR, and enables the parity error input, ERR. This bit is valid only when UnbuffDimm=0. BIOS should program this to 1 for registered DIMM systems.															
7	<b>DramDrvWeak: DRAM drivers weak mode.</b> Read-write. This specifies the programming of the DRAM data drive strength mode when the EMRS command is issued during DRAM initialization (F2x[1, 0]90[InitDram]). 1=Weak drive strength mode. 0=Normal drive strength mode. This bit is undefined when F2x[1, 0]94[Ddr3Mode]=1.															
6	<b>DisDqsBar: disable low differential DQS pin.</b> Read write. This specifies the programming of the DRAM low-DQS (of the differential pairs) signal enable when the EMRS command is issued during DDR2 DRAM initialization (F2x[1, 0]90[InitDram]). 1=Disable low DQS pins. 0=Enable low DQS pins. This bit is undefined when F2x[1, 0]94[Ddr3Mode]=1.															

5:4	<p><b>DramTerm: DRAM termination.</b> Read-write. This specifies the programming of the DRAM termination value (Rtt) when the EMRS command is issued during DDR2 DRAM initialization (F2x[1, 0]90[InitDram]). This field is undefined when F2x[1, 0]94[Ddr3Mode]=1. For DDR3 termination values, see F2x[1, 0]84[DramTerm].</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>DDR2 definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>On die termination disabled.</td> </tr> <tr> <td>01b</td> <td>75 ohms.</td> </tr> <tr> <td>10b</td> <td>150 ohms.</td> </tr> <tr> <td>11b</td> <td>50 ohms.</td> </tr> </tbody> </table>	<u>Bits</u>	<u>DDR2 definition</u>	00b	On die termination disabled.	01b	75 ohms.	10b	150 ohms.	11b	50 ohms.																				
<u>Bits</u>	<u>DDR2 definition</u>																														
00b	On die termination disabled.																														
01b	75 ohms.																														
10b	150 ohms.																														
11b	50 ohms.																														
3:2	<p><b>PllLockTime: registered DIMM PLL lock time.</b> Read-write. This specifies registered DIMM PLL lock time as follows when F2x[1, 0]94[Ddr3Mode]=0:</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>DDR Rate</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>DDR2-800</td> <td>15us</td> </tr> <tr> <td>00b</td> <td>DDR2-667</td> <td>18us</td> </tr> <tr> <td>00b</td> <td>DDR2-533</td> <td>22.5us</td> </tr> <tr> <td>00b</td> <td>DDR2-400</td> <td>30us</td> </tr> <tr> <td>01b</td> <td>DDR2-800</td> <td>6us</td> </tr> <tr> <td>01b</td> <td>DDR2-667</td> <td>7.2us</td> </tr> <tr> <td>01b</td> <td>DDR2-533</td> <td>9us</td> </tr> <tr> <td>01b</td> <td>DDR2-400</td> <td>12us</td> </tr> <tr> <td>1xb</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	<u>Bits</u>	<u>DDR Rate</u>	<u>Definition</u>	00b	DDR2-800	15us	00b	DDR2-667	18us	00b	DDR2-533	22.5us	00b	DDR2-400	30us	01b	DDR2-800	6us	01b	DDR2-667	7.2us	01b	DDR2-533	9us	01b	DDR2-400	12us	1xb	Reserved	
<u>Bits</u>	<u>DDR Rate</u>	<u>Definition</u>																													
00b	DDR2-800	15us																													
00b	DDR2-667	18us																													
00b	DDR2-533	22.5us																													
00b	DDR2-400	30us																													
01b	DDR2-800	6us																													
01b	DDR2-667	7.2us																													
01b	DDR2-533	9us																													
01b	DDR2-400	12us																													
1xb	Reserved																														
1	<p><b>ExitSelfRef: exit self refresh (after suspend to RAM or for DRAM training) command.</b> Read, write-1-only. Writing a 1 to this bit causes the DRAM controller to bring the DRAMs out of self refresh mode. 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. Note: this bit should not be set if the DCT is disabled.</p>																														
0	<p><b>InitDram: initialize DRAM.</b> Read, write-1-only. Writing a 1 to this bit causes the DRAM controller to execute the DRAM initialization sequence described by the JEDEC specification. This command should be executed by BIOS when booting from an unpowered state (ACPI S4, S5 or G3; not S3, suspend to RAM), after the DRAM controller configuration registers are properly initialized. This bit is read as a 1 while the DRAM initialization sequence is executing; it is read as 0 at all other times. When this bit is written to a 1, the new value of the other fields in this register that are updated concurrently are used in the initialization sequence. BIOS must wait until the bit is 0 before accessing any other register in the DRAM controller or any memory attached to the controller. BIOS must not set this bit on a DCT with no attached DIMMs. See 2.8.9.6 [DRAM Device and Controller Initialization] for more details.</p>																														

### F2x[1, 0]94 DRAM Configuration High Register

Reset: 0008 0200h. See 2.8.1 [DCT Configuration Registers] for general programming information about DCT configuration registers.

Bits	Description																												
31:28	<p><b>FourActWindow[3:0]: four bank activate window.</b> Read-write. FourActWindow specifies the rolling tFAW window during which no more than 4 banks in an 8-bank device are activated, per JEDEC DDR2 and DDR3 specifications. The meaning of FourActWindow varies with F2x[1, 0]94[Ddr3Mode], and when F2x[1, 0]94[MemClkFreq]=100b to support DDR2-1066 as follows:</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>DDR2 window size</th> <th>Bits</th> <th>DDR2-1066/DDR3 window size</th> </tr> </thead> <tbody> <tr> <td>0000b</td> <td>No tFAW window restriction.</td> <td>0000b</td> <td>No tFAW window restriction.</td> </tr> <tr> <td>0001b</td> <td>8 MEMCLK cycles.</td> <td>0001b</td> <td>16 MEMCLK cycles.</td> </tr> <tr> <td>0010b</td> <td>9 MEMCLK cycles.</td> <td>0010b</td> <td>18 MEMCLK cycles.</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>1101b</td> <td>20 MEMCLK cycles.</td> <td>1001b</td> <td>32 MEMCLK cycles.</td> </tr> <tr> <td>1110b - 1111b</td> <td>Reserved</td> <td>1010b - 1111b</td> <td>Reserved</td> </tr> </tbody> </table> <p>See 2.8.9.5.6 [FourActWindow (Four Bank Activate Window or tFAW)] for information on how to program this field.</p>	Bits	DDR2 window size	Bits	DDR2-1066/DDR3 window size	0000b	No tFAW window restriction.	0000b	No tFAW window restriction.	0001b	8 MEMCLK cycles.	0001b	16 MEMCLK cycles.	0010b	9 MEMCLK cycles.	0010b	18 MEMCLK cycles.	...	...	...	...	1101b	20 MEMCLK cycles.	1001b	32 MEMCLK cycles.	1110b - 1111b	Reserved	1010b - 1111b	Reserved
Bits	DDR2 window size	Bits	DDR2-1066/DDR3 window size																										
0000b	No tFAW window restriction.	0000b	No tFAW window restriction.																										
0001b	8 MEMCLK cycles.	0001b	16 MEMCLK cycles.																										
0010b	9 MEMCLK cycles.	0010b	18 MEMCLK cycles.																										
...	...	...	...																										
1101b	20 MEMCLK cycles.	1001b	32 MEMCLK cycles.																										
1110b - 1111b	Reserved	1010b - 1111b	Reserved																										
27:24	<p><b>DcqBypassMax: DRAM controller queue bypass maximum.</b> Read-write. 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. For optimal performance, it is recommended that this field be programmed to Fh.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>No bypass; the oldest request is never bypassed.</td> </tr> <tr> <td>1h</td> <td>The oldest request may be bypassed no more than 1 time.</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>Fh</td> <td>The oldest request may be bypassed no more than 15 times.</td> </tr> </tbody> </table>	Bits	Definition	0h	No bypass; the oldest request is never bypassed.	1h	The oldest request may be bypassed no more than 1 time.	...	...	Fh	The oldest request may be bypassed no more than 15 times.																		
Bits	Definition																												
0h	No bypass; the oldest request is never bypassed.																												
1h	The oldest request may be bypassed no more than 1 time.																												
...	...																												
Fh	The oldest request may be bypassed no more than 15 times.																												
23	<p><b>ProcOdtDis: processor on-die termination disable.</b> Read-write. 1=The processor-side on-die termination is disabled. 0=Processor-side on-die termination enabled. See F2x[1, 0]9C_x00[ProcOdt] for ODT definitions. Changes to this bit must be done prior to DRAM initialization.</p>																												
22	<p><b>BankSwizzleMode: bank swizzle mode.</b> Read-write. 1=Remaps the DRAM device bank address bits as a function of normalized physical address bits. Each of the bank address bits, as specified in Table 85 and Table 86 of F2x[1, 0]80, are remapped as follows:</p> <p>Define X as a bank address bit (e.g., X=15 if the bank bit is specified to be address bit 15).  <math>X' = X</math>, if the DCT is in 64-bit ungang mode, or <math>X+1</math> in 128-bit gang mode.  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' + 2) \wedge S(X' + 4)</math>; for a 4-bank DRAM.  <math>B = S(X') \wedge S(X' + 3) \wedge S(X' + 6)</math>; for an 8-bank DRAM.</p> <p>For example, encoding 02h of Table 85 would be remapped from bank[1:0]={A14, A13} to the following for a 64-bit DCT: Bank[1:0] = {A14 ^ A16 ^ A18, A13 ^ A15 ^ A17}.  For example, if [18:13]=110001b, then Bank[1:0] = {0 ^ 0 ^ 1, 1 ^ 0 ^ 1} = {1, 0}.</p> <p>BIOS should set this bit to 1b.</p>																												
21	<p><b>FreqChgInProg: frequency change in progress.</b> Read-only. 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	<b>SlowAccessMode: slow access mode (a.k.a. 2T mode).</b> Read-write. 1=One additional MEMCLK of setup time is provided on all DRAM address and control signals (not including CS, CKE, and ODT); i.e., these signals are driven for two MEMCLK cycles rather than one. 0=DRAM address and control signals are driven for one MEMCLK cycle. 2T mode may be needed in order to meet electrical requirements of certain DIMM speed and loading configurations.						
19	<b>DcqArbBypassEn: DRAM controller arbiter bypass enable.</b> Revision B and later: Read-write. 1=Bypass the arbitration logic when there is only one entry in the DRAM controller queue. 0=DCQ entries are always passed through the arbiter.						
18	<b>FourRankRDimm: four rank registered DIMM connected.</b> Read-write. 1=Four-rank registered DIMMs are connected to the channel. In this mode, only two DIMMs per channel are supported. See <a href="#">F2x[1, 0][5C:40]</a> for configuration information in this mode.						
17	Reserved.						
16	<p><b>PowerDownMode: power down mode.</b> Read-write. This specifies how a DIMM or group of DIMMs enters power down mode when enabled by <a href="#">F2x[1, 0]94[PowerDownEn]</a>. A DIMM enters power down mode when the DCT deasserts the CKE pin to that DIMM. The command and address signals tristate one MEMCLK after CKE deasserts. There are two CKE pins per DRAM channel. For each channel:</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0b</td> <td> <p>Channel CKE control mode. The DRAM channel is placed in power down mode when all chip selects associated with the channel are idle. Both CKE pins for the channel operate in lock step, in terms of placing the channel DIMMs in power down mode.</p> <p>For DDR2 DIMMs:</p> <ul style="list-style-type: none"> <li>- CKE0 is connected to CS0, CS1, CS4, CS5 (the even DIMMs);</li> <li>- CKE1 is connected to CS2, CS3, CS6, CS7 (the odd DIMMs).</li> </ul> <p>For DDR3 DIMMs:</p> <ul style="list-style-type: none"> <li>- CKE0 is connected to CS0, CS2, CS4, CS6 (the even ranks);</li> <li>- CKE1 is connected to CS1, CS3, CS5, CS7 (the odd ranks).</li> </ul> </td> </tr> <tr> <td>1b</td> <td> <p>Chip select CKE control mode. A chip select is placed in power down mode when no transactions are pending for the chip select. This mode is expected to be used in systems with one DIMM per channel.</p> <ul style="list-style-type: none"> <li>- CKE0 is associated with CS0.</li> <li>- CKE1 is associated with CS1 .</li> </ul> </td> </tr> </tbody> </table> <p>This bit interacts with <a href="#">F2x[1, 0]84[PchgPDMoSel]</a> as follows:</p> <ul style="list-style-type: none"> <li>• If PchgPDMoSel=0, then the DCT waits for both ranks of a dual rank DIMM to become idle before issuing commands to go into deep (DLL off) power down.</li> <li>• If PchgPDMoSel=1 then an idle rank is placed in shallow (DLL on) power down independent of the activity to another rank and if both ranks are idle for 256 clocks, the DCT wakes them and prepares them for deep power down mode.</li> </ul> <p>In configurations where both modes are valid, BIOS should program this to 1.</p>	Bit	Description	0b	<p>Channel CKE control mode. The DRAM channel is placed in power down mode when all chip selects associated with the channel are idle. Both CKE pins for the channel operate in lock step, in terms of placing the channel DIMMs in power down mode.</p> <p>For DDR2 DIMMs:</p> <ul style="list-style-type: none"> <li>- CKE0 is connected to CS0, CS1, CS4, CS5 (the even DIMMs);</li> <li>- CKE1 is connected to CS2, CS3, CS6, CS7 (the odd DIMMs).</li> </ul> <p>For DDR3 DIMMs:</p> <ul style="list-style-type: none"> <li>- CKE0 is connected to CS0, CS2, CS4, CS6 (the even ranks);</li> <li>- CKE1 is connected to CS1, CS3, CS5, CS7 (the odd ranks).</li> </ul>	1b	<p>Chip select CKE control mode. A chip select is placed in power down mode when no transactions are pending for the chip select. This mode is expected to be used in systems with one DIMM per channel.</p> <ul style="list-style-type: none"> <li>- CKE0 is associated with CS0.</li> <li>- CKE1 is associated with CS1 .</li> </ul>
Bit	Description						
0b	<p>Channel CKE control mode. The DRAM channel is placed in power down mode when all chip selects associated with the channel are idle. Both CKE pins for the channel operate in lock step, in terms of placing the channel DIMMs in power down mode.</p> <p>For DDR2 DIMMs:</p> <ul style="list-style-type: none"> <li>- CKE0 is connected to CS0, CS1, CS4, CS5 (the even DIMMs);</li> <li>- CKE1 is connected to CS2, CS3, CS6, CS7 (the odd DIMMs).</li> </ul> <p>For DDR3 DIMMs:</p> <ul style="list-style-type: none"> <li>- CKE0 is connected to CS0, CS2, CS4, CS6 (the even ranks);</li> <li>- CKE1 is connected to CS1, CS3, CS5, CS7 (the odd ranks).</li> </ul>						
1b	<p>Chip select CKE control mode. A chip select is placed in power down mode when no transactions are pending for the chip select. This mode is expected to be used in systems with one DIMM per channel.</p> <ul style="list-style-type: none"> <li>- CKE0 is associated with CS0.</li> <li>- CKE1 is associated with CS1 .</li> </ul>						
15	<b>PowerDownEn: power down mode enable.</b> Read-write. 1=Power down mode is enabled. When in power down mode, if all pages of the DRAMs associated with a CKE pin are closed, then these parts are placed in power down mode. Only pre-charge power down mode is supported, not active power down mode. BIOS should program this to 1 for unbuffered and SODIMM systems. For registered DIMM systems, PowerDownEn may be set or cleared based on power and performance tradeoffs.						

14	<b>DisDramInterface: disable the DRAM interface.</b> Read-write. 1=The DRAM controller is disabled and the DRAM phy is placed into a low power state. This bit must be set if there are no DIMMs connected to the DCT. BIOS must set <a href="#">F2x[1, 0]9C_x0C[CKETri, ODTTri, and ChipSelTri]</a> for the DCT a minimum of 24 MEMCLKs before setting DisDramInterface in order to tri-state those DRAM interface pins. Note if this bit is set, <a href="#">F2x[1, 0]90[DimmEccEn]</a> and <a href="#">F2x[1, 0]90[ParEn]</a> should not be set to avoid spurious MCA errors.										
13	<b>DisSimulRdWr: disable simultaneous read and write.</b> Read-write. 1=Disable the possibility of simultaneous reads from one DCT and writes to the other DCT. If data-integrity issues result from the additional electrical noise present when simultaneous read and write activity occurs, then this bit may be set at the cost of some amount of performance. This bit should always be low if the DCTs are ganged ( <a href="#">F2x110[DctGangEn]</a> ). The value of this bit should be programmed the same for both DCTs.										
12	<b>RDqsEn: read DQS enable.</b> Read-write. This is applied to the DRAM device's DDR2-defined EMRS(1) or DDR3-defined MR1 registers during DRAM initialization (see <a href="#">2.8.9.6 [DRAM Device and Controller Initialization]</a> ). This RDQS/TDQS DRAM function should only be set for x8 registered DIMMs when x4 and x8 registered DIMMs are mixed on a channel. The definition of this bit varies with <a href="#">F2x[1, 0]94[Ddr3Mode]</a> as follows: <u>DDR2 definition:</u> 1=The DIMM DM pins are used as read DQS pins and data masking is disabled. 0=DM pins function as data mask pins. <u>DDR3 definition:</u> 1=The DIMM DM and DQS#[17:9] pins are used to provide DQS termination of x8 based DIMMs for accesses to x4 based DIMMs and data masking is disabled. 0=DM pins function as data mask pins.										
11:10	<b>ZqcsInterval: ZQ calibration short interval.</b> Read-write. This field specifies the programmable interval for the controller to send out the DRAM ZQ calibration short command. ZqcsInterval is defined only when <a href="#">F2x[1, 0]94[Ddr3Mode] = 1</a> . See <a href="#">2.8.9.9.1 [Write Levelization Training]</a> . <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</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 (recommended)</td> </tr> <tr> <td>11b</td> <td>256 ms</td> </tr> </tbody> </table>	Bits	Definition	00b	ZQ calibration short command is disabled	01b	64 ms	10b	128 ms (recommended)	11b	256 ms
Bits	Definition										
00b	ZQ calibration short command is disabled										
01b	64 ms										
10b	128 ms (recommended)										
11b	256 ms										
9	<b>LegacyBiosMode.</b> Read-write. 0=Normal DCT functionality. 1=Legacy BIOS mode is enabled and DCT1 is disabled. In this mode, <a href="#">[The DRAM DQS Receiver Enable Timing Control Registers] F2x[1, 0]9C_x[2B:10]</a> value in <a href="#">DqsRcvEnGrossDelay[0]</a> is programmed into the register fields of <a href="#">DqsRcvEnGrossDelay[1, 2, 3, 4, 5, 6, 7]</a> and <a href="#">DqsRcvEnGrossDelayCheck</a> for each corresponding DIMM (as defined by <a href="#">[The DRAM CS Base Address Registers] F2x[1, 0][5C:40]</a> ). The value in the field <a href="#">DqsRcvEnFineDelay[0]</a> is programmed into the register fields of <a href="#">DqsRcvEnFineDelay[1, 2, 3, 4, 5, 6, 7]</a> and <a href="#">DqsRcvEnFineDelayCheck</a> . While this bit is set, MemClkFreq is fixed at 200 MHz and accesses to registers <a href="#">F2x[1, 0]9C_x[3:0]0[2:1]</a> , <a href="#">F2x[1, 0]9C_x[3:0]03</a> , <a href="#">F2x[1, 0]9C_x[3:0]0[6:5]</a> , <a href="#">F2x[1, 0]9C_x[3:0]07</a> , and <a href="#">F2x[1, 0]9C_x[4A:30]</a> are blocked. This bit is used to maintain BIOS compatibility with the NPT Family 0Fh memory controller. Family 10h BIOS should clear this bit before programming <a href="#">F2x[1, 0]9C_x[2B:10]</a> .										
8	<b>Ddr3Mode.</b> Read-write or read-only based on product. This field is read-only if the processor does not support DDR3. This bit must be set by BIOS based on the types of DIMMs connected to the DCT. 0=DDR2 mode. 1=DDR3 mode. Both DCTs must be programmed to the same DIMM type.										
7:4	Reserved.										



3	<b>MemClkFreqVal: memory clock frequency valid.</b> Read-write. System BIOS should set this bit after setting up <a href="#">F2x[1, 0]94[MemClkFreq]</a> to the proper value. This indicates to the DRAM controller that it may start driving MEMCLK at the proper frequency. BIOS should poll <a href="#">FreqChgInProg</a> to determine when the DRAM-interface clocks are stable. Note: this bit should not be set if the DCT is disabled. BIOS must change each DCT's operating frequency in order. See <a href="#">2.8.9.6 [DRAM Device and Controller Initialization]</a> .																											
2:0	<p><b>MemClkFreq: memory clock frequency.</b> Read-write. This field specifies the frequency of the DRAM interface (MEMCLK). The definition varies with the DDR type, <a href="#">F2x[1, 0]94[Ddr3Mode]</a>.</p> <table border="1" data-bbox="280 478 974 791"> <thead> <tr> <th>Bits</th> <th>DDR2 Definition</th> <th>DDR3 Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>200 MHz</td> <td>Reserved</td> </tr> <tr> <td>001b</td> <td>266 MHz</td> <td>Reserved</td> </tr> <tr> <td>010b</td> <td>333 MHz</td> <td>333 MHz revision D only.</td> </tr> <tr> <td>011b</td> <td>400 MHz</td> <td>400 MHz</td> </tr> <tr> <td>100b</td> <td>533 MHz</td> <td>533 MHz</td> </tr> <tr> <td>101b</td> <td>Reserved</td> <td>667 MHz</td> </tr> <tr> <td>110b</td> <td>Reserved</td> <td>800 MHz</td> </tr> <tr> <td>111b</td> <td>Reserved</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	DDR2 Definition	DDR3 Definition	000b	200 MHz	Reserved	001b	266 MHz	Reserved	010b	333 MHz	333 MHz revision D only.	011b	400 MHz	400 MHz	100b	533 MHz	533 MHz	101b	Reserved	667 MHz	110b	Reserved	800 MHz	111b	Reserved	Reserved
Bits	DDR2 Definition	DDR3 Definition																										
000b	200 MHz	Reserved																										
001b	266 MHz	Reserved																										
010b	333 MHz	333 MHz revision D only.																										
011b	400 MHz	400 MHz																										
100b	533 MHz	533 MHz																										
101b	Reserved	667 MHz																										
110b	Reserved	800 MHz																										
111b	Reserved	Reserved																										

### **F2x[1, 0]98 DRAM Controller Additional Data Offset Register**

Reset: 8000 0000h. The DCTs each include an array of registers called [F2x\[1, 0\]9C\\_x\[107:00\]](#), which are defined following [F2x\[1, 0\]9C](#). These are used primarily to control DRAM-interface electrical parameters. [\[The DRAM Controller Additional Data Offset Register\] F2x\[1, 0\]98](#) and [\[The DRAM Controller Additional Data Port\] F2x\[1, 0\]9C](#) are used to access [F2x\[1, 0\]9C\\_x\[107:00\]](#). The register number (i.e., the number that follows “\_x” in the register mnemonic) is specified by [F2x\[1, 0\]98\[DctOffset\]](#). Access to these registers is accomplished as follows:

- Reads:
  - Write the register number to [F2x\[1, 0\]98\[DctOffset\]](#) with [F2x\[1, 0\]98\[DctAccessWrite\]=0](#).
  - Poll [F2x\[1, 0\]98\[DctAccessDone\]](#) until it is high.
  - Read the register contents from [F2x\[1, 0\]9C](#).
- Writes:
  - Write all 32 bits to the register data to [F2x\[1, 0\]9C](#) (individual byte writes are not supported).
  - Write the register number to [F2x\[1, 0\]98\[DctOffset\]](#) with [F2x\[1, 0\]98\[DctAccessWrite\]=1](#).
  - Poll [F2x\[1, 0\]98\[DctAccessDone\]](#) until it is high to ensure that the contents of the write have been delivered to the phy.

Writes to any register in this additional address space causes the FIFO pointers to be reset.

See [2.8.1 \[DCT Configuration Registers\]](#) for general programming information about DCT configuration registers. Note, however, that [F2x198](#), [F2x098](#), [F2x19C\\_x\[107:00\]](#), and [F2x09C\\_x\[107:00\]](#), may all be programmed to different values even if the DCTs are in ganged mode.



Bits	Description
31	<b>DctAccessDone: DRAM controller access done.</b> Read-only. 1=The access to one of the F2x[1, 0]9C_x[107:00] registers is complete. 0=The access is still in progress. The hardware also clears and sets this field automatically when any of the following bits are programmed to 1 by BIOS: <ul style="list-style-type: none"> <li>• F2x[1, 0]7C[EnDramInit].</li> <li>• F2x[1, 0]90[InitDram].</li> <li>• F2x[1, 0]90[ExitSelfRef].</li> <li>• F2x[1, 0]94[MemClkFreqVal] if the controller has already been initialized with one of the other three bits.</li> </ul>
30	<b>DctAccessWrite: DRAM controller read/write select.</b> Write-only. 0=Read one of the F2x[1, 0]9C_x[107:00] registers. 1=Write one of the F2x[1, 0]9C_x[107:00] registers.
29:0	<b>DctOffset: DRAM controller offset.</b> Read-write.

### F2x[1, 0]9C DRAM Controller Additional Data Port

See F2x[1, 0]98 for details about this port.

### F2x[1, 0]9C\_x00 DRAM Output Driver Compensation Control Register

Reset: see field definitions. See F2x[1, 0]98 for register access information. See 2.8.9.5.8 [DRAM Address Timing and Output Driver Compensation Control] for information on how to program this register.

Bits	Description																																	
31:30	Reserved.																																	
29:28	<b>ProcOdt: processor on-die termination.</b> Read-write. Cold reset: 00b. This field specifies the resistance of the on-die termination resistors. The definition varies with the DDR type specified by F2x[1, 0]94[Ddr3Mode]. This field is valid only when F2x[1, 0]94[ProcOdtDis]=0. See F2x[1,0]9C_x0D0F0[F,8:0]0[8,4,0][OdtModeInvert]. <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 20px;"><u>Bits</u></td> <td style="padding-right: 20px;"><u>DDR2 for Revision B, RB-C2, BL-C2</u></td> <td><u>DDR2 for Revision DA-C, C3 and later</u></td> </tr> <tr> <td>00b</td> <td>300 ohms +/- 20%</td> <td>320 ohms +/- 10%</td> </tr> <tr> <td>01b</td> <td>150 ohms +/- 20%</td> <td>160 ohms +/- 10%</td> </tr> <tr> <td>10b</td> <td>75 ohms +/- 20%</td> <td>107 ohms +/- 10%</td> </tr> <tr> <td>11b</td> <td>Reserved</td> <td>80 ohms +/- 10%</td> </tr> <tr> <td colspan="3"> </td> </tr> <tr> <td><u>Bits</u></td> <td><u>DDR3</u></td> <td></td> </tr> <tr> <td>00b</td> <td>240 ohms +/- 10%</td> <td></td> </tr> <tr> <td>01b</td> <td>120 ohms +/- 10%</td> <td></td> </tr> <tr> <td>10b</td> <td>60 ohms +/- 10%</td> <td></td> </tr> <tr> <td>11b</td> <td>Reserved</td> <td></td> </tr> </table>	<u>Bits</u>	<u>DDR2 for Revision B, RB-C2, BL-C2</u>	<u>DDR2 for Revision DA-C, C3 and later</u>	00b	300 ohms +/- 20%	320 ohms +/- 10%	01b	150 ohms +/- 20%	160 ohms +/- 10%	10b	75 ohms +/- 20%	107 ohms +/- 10%	11b	Reserved	80 ohms +/- 10%	 			<u>Bits</u>	<u>DDR3</u>		00b	240 ohms +/- 10%		01b	120 ohms +/- 10%		10b	60 ohms +/- 10%		11b	Reserved	
<u>Bits</u>	<u>DDR2 for Revision B, RB-C2, BL-C2</u>	<u>DDR2 for Revision DA-C, C3 and later</u>																																
00b	300 ohms +/- 20%	320 ohms +/- 10%																																
01b	150 ohms +/- 20%	160 ohms +/- 10%																																
10b	75 ohms +/- 20%	107 ohms +/- 10%																																
11b	Reserved	80 ohms +/- 10%																																
<u>Bits</u>	<u>DDR3</u>																																	
00b	240 ohms +/- 10%																																	
01b	120 ohms +/- 10%																																	
10b	60 ohms +/- 10%																																	
11b	Reserved																																	
27:22	Reserved.																																	

21:20	<p><b>DqsDrvStren: DQS drive strength.</b> Read-write. Cold reset: 11b. This field specifies the drive strength of the DQS pins.</p> <table> <tr> <td>00b</td> <td>0.75x</td> <td>10b</td> <td>1.25x</td> </tr> <tr> <td>01b</td> <td>1.0x</td> <td>11b</td> <td>1.5x</td> </tr> </table> <p>Note: the DM[8:0] and DQS[17:9] functions share pins on the DIMM connector. The function selection is applied based on whether the DIMM is populated with by-4 (x4) DRAM devices, in which case the DQS[17:9] function is applied, or not (x8 or x16 DRAM devices), in which case the DM[8:0] function is applied. However, the DM function is associated with the data pin group and should therefore be controlled DataDrvStren. While the processor supports concurrent population of x4 and non-x4 DIMMs, the determination as to which field controls the drive strength of these pins is applied statically based on these rules:</p> <ul style="list-style-type: none"> <li>• If all DIMMs of an ungangned channel are populated with non-x4 devices, DataDrvStren is applied.</li> <li>• If the channels are gangned and populated with all non-x4 devices, DataDrvStren is applied.</li> <li>• If any DIMMs of an ungangned channel are populated with x4 devices, DqsDrvStren is applied.</li> <li>• If the channels are gangned and populated with any x4 devices, DqsDrvStren is applied.</li> </ul>	00b	0.75x	10b	1.25x	01b	1.0x	11b	1.5x
00b	0.75x	10b	1.25x						
01b	1.0x	11b	1.5x						
19:18	Reserved.								
17:16	<p><b>DataDrvStren: data drive strength.</b> Read-write. Cold reset: 11b. This field specifies the drive strength of the DRAM data pins.</p> <table> <tr> <td>00b</td> <td>0.75x.</td> <td>10b</td> <td>1.25x</td> </tr> <tr> <td>01b</td> <td>1.0x</td> <td>11b</td> <td>1.5x</td> </tr> </table> <p>See the note in DqsDrvStren regarding how this field may be applied to DM signals as well.</p>	00b	0.75x.	10b	1.25x	01b	1.0x	11b	1.5x
00b	0.75x.	10b	1.25x						
01b	1.0x	11b	1.5x						
15:14	Reserved.								
13:12	<p><b>ClkDrvStren: MEMCLK drive strength.</b> Read-write. Cold reset: 11b. This field specifies the drive strength of the MEMCLK pins.</p> <table> <tr> <td>00b</td> <td>0.75x.</td> <td>10b</td> <td>1.25x</td> </tr> <tr> <td>01b</td> <td>1.0x</td> <td>11b</td> <td>1.5x</td> </tr> </table>	00b	0.75x.	10b	1.25x	01b	1.0x	11b	1.5x
00b	0.75x.	10b	1.25x						
01b	1.0x	11b	1.5x						
11:10	Reserved.								
9:8	<p><b>AddrCmdDrvStren: address/command drive strength.</b> Read-write. Cold reset: 11b. This field specifies the drive strength of the address, RAS, CAS, WE, bank and parity pins.</p> <table> <tr> <td>00b</td> <td>1.0x.</td> <td>10b</td> <td>1.5x</td> </tr> <tr> <td>01b</td> <td>1.25x</td> <td>11b</td> <td>2.0x</td> </tr> </table>	00b	1.0x.	10b	1.5x	01b	1.25x	11b	2.0x
00b	1.0x.	10b	1.5x						
01b	1.25x	11b	2.0x						
7:6	Reserved.								
5:4	<p><b>CsOdtDrvStren: CS/ODT drive strength.</b> Read-write. Cold reset: 11b. This field specifies the drive strength of the CS and ODT pins.</p> <table> <tr> <td>00b</td> <td>1.0x.</td> <td>10b</td> <td>1.5x</td> </tr> <tr> <td>01b</td> <td>1.25x</td> <td>11b</td> <td>2.0x</td> </tr> </table>	00b	1.0x.	10b	1.5x	01b	1.25x	11b	2.0x
00b	1.0x.	10b	1.5x						
01b	1.25x	11b	2.0x						
3:2	Reserved.								
1:0	<p><b>CkeDrvStren: CKE drive strength.</b> Read-write. Cold reset: 11b. This field specifies the drive strength of the CKE pins.</p> <table> <tr> <td>00b</td> <td>1.0x.</td> <td>10b</td> <td>1.5x</td> </tr> <tr> <td>01b</td> <td>1.25x</td> <td>11b</td> <td>2.0x</td> </tr> </table>	00b	1.0x.	10b	1.5x	01b	1.25x	11b	2.0x
00b	1.0x.	10b	1.5x						
01b	1.25x	11b	2.0x						

### **F2x[1, 0]9C\_x[3:0]0[2:1] DRAM Write Data Timing [High:Low] Registers**

See F2x[1, 0]98 for register access information.

The following register list is expanded as follows:

<u>DctOffset</u>	<u>Register</u>
0000_0001h	DRAM Write Data Timing Low DIMM 0: (bytes 0,1,2,3)
0000_0002h	DRAM Write Data Timing High DIMM 0: (bytes 4,5,6,7)
0000_0003h	DRAM Write ECC Timing DIMM 0
0000_0101h	DRAM Write Data Timing Low DIMM 1: (bytes 0,1,2,3)
0000_0102h	DRAM Write Data Timing High DIMM 1: (bytes 4,5,6,7)
0000_0103h	DRAM Write ECC Timing DIMM 1
0000_0201h	DRAM Write Data Timing Low DIMM 2: (bytes 0,1,2,3)
0000_0202h	DRAM Write Data Timing High DIMM 2: (bytes 4,5,6,7)
0000_0203h	DRAM Write ECC Timing DIMM 2
0000_0301h	DRAM Write Data Timing Low DIMM 3: (bytes 0,1,2,3)
0000_0302h	DRAM Write Data Timing High DIMM 3: (bytes 4,5,6,7)
0000_0303h	DRAM Write ECC Timing DIMM 3

These registers control the timing of write data 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 2.8.9.9 [DRAM Training] for information on how to use these registers. To ensure unique values are written to each timing control register, BIOS must program these registers in consecutive DIMM order; i.e., program DIMM 0 register values first followed by the DIMM 1 values, etc.

Fine timing (WrDatFineDlyByte):

Delay = WrDatFineDlyByte \* 1/64 of a MEMCLK, ranging from 0/64 to 31/64 MEMCLKs.

Gross timing (WrDatGrossDlyByte):

00b	No delay
01b	0.5 MEMCLK delay
10b	1.0 MEMCLK delay
11b	1.5 MEMCLK delay

Note: WrDatGrossDlyByte should be zero for DDR2 DIMMs. For revision B, WrDatGrossDlyByte is reserved for registers where DctOffSet is 2XXh or 3XXh.

The total delay is the sum of these two fields, ranging from 0 to 1 and 63/64 MEMCLKs.

Bits	Description
31	Reserved.
30:29	<b>WrDatGrossDlyByte[7, 3]: write data gross delay byte[7, 3].</b> Read-write. Reset: 0.
28:24	<b>WrDatFineDlyByte[7, 3]: write data fine delay byte[7, 3].</b> Read-write. Cold reset: 0Fh.
23	Reserved.
22:21	<b>WrDatGrossDlyByte[6, 2]: write data gross delay byte[6, 2].</b> Read-write. Reset: 0.
20:16	<b>WrDatFineDlyByte[6, 2]: write data fine delay byte[6, 2].</b> Read-write. Cold reset: 0Fh.
15	Reserved.
14:13	<b>WrDatGrossDlyByte[5, 1]: write data gross delay byte[5, 1].</b> Read-write. Reset: 0.
12:8	<b>WrDatFineDlyByte[5, 1]: write data fine delay byte[5, 1].</b> Read-write. Cold reset: 0Fh.
7	Reserved.
6:5	<b>WrDatGrossDlyByte[4, 0]: write data gross delay byte[4, 0].</b> Read-write. Reset: 0.
4:0	<b>WrDatFineDlyByte[4, 0]: write data fine delay byte[4, 0].</b> Read-write. Cold reset: 0Fh.

**F2x[1, 0]9C\_x[3:0]03 DRAM Write ECC Timing Register**

See F2x[1, 0]98 for register access information. See F2x[1, 0]9C\_x[3:0]0[2:1] for a partially expanded list including DRAM Write ECC Timing Registers and for revision specific information.

These registers specify the delay that is added to the ECC write data bits with respect to MEMCLK. The delay starts 1 UI before the rising edge of MEMCLK corresponding to the CAS-write-latency. The total delay is the sum of the fields, ranging from 0 to 1 and 31/64 MEMCLKs. See 2.8.9.9 [DRAM Training] for information on how to use this register.

Bits	Description
31:7	Reserved.
6:5	<b>WrChkGrossDly: write data ECC gross delay.</b> Read-write. Reset: 0. This is encoded as follows: 00b No delay. 01b 0.5 MEMCLK delay. 10b 1.0 MEMCLK delay. 11b 1.5 MEMCLK delay.
4:0	<b>WrChkFineDly: write data ECC fine delay.</b> Read-write. Cold reset: 0Fh. This is encoded as follows: Delay = WrChkFineDly * 1/64 of a MEMCLK, ranging from 0/64 to 31/64 MEMCLKs.

**F2x[1, 0]9C\_x04 DRAM Address/Command Timing Control Register**

See F2x[1, 0]98 for register access information. This register controls the timing of the address, command, chip select, ODT and clock enable pins with respect to MEMCLK. See the figure below. This register is used to adjust both the setup and hold time at the DIMM. It is recommended that the address and commands are launched 3/4 of a cycle ahead of the rising edge of MEMCLK. See 2.8.9.5.8 [DRAM Address Timing and Output Driver Compensation Control] for information on how to program this register. When programming this register, F2x[1, 0]9C\_x0C should be written prior to writing F2x[1, 0]9C\_x04.

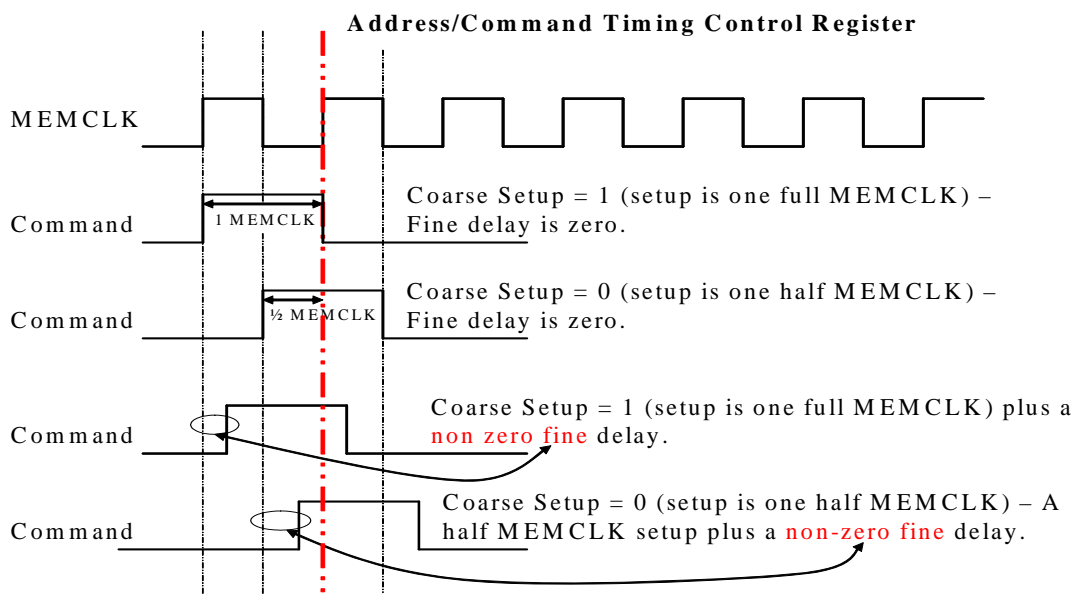


Figure 17: Address/Command Timing at the Processor Pins

2T timing is controlled by `F2x[1, 0]94[SlowAccessMode]`. Note: if the DCT channels are ganged (see `F2x110[DctGangEn]`), then this register must be programmed before setting `F2x[1, 0]94[MemClkFreqVal]=1`. Note: if a setup time (course delay) field is changed and `F2x[1, 0]94[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. This bit selects the default setup time for the address and command pins versus MEMCLK. 0b 1/2 MEMCLK (1 1/2 MEMCLK for 2T timing) 1b 1 MEMCLK (2 MEMCLKs for 2T timing)
20:16	<b>AddrCmdFineDelay: address/command fine delay.</b> Read-write. Cold reset: 00h. This field specifies the time that the address and command pins are delayed from the default setup time. 0_0000b No delay 0_0001b 1/64 MEMCLK delay 0_0010b 2/64 MEMCLK delay ... 1_1111b 31/64 MEMCLK delay
15:14	Reserved.
13	<b>CsOdtSetup: CS/ODT setup time.</b> Read-write. Reset: 0. This bit selects the default setup time for the CS and ODT pins versus MEMCLK. 0b 1/2 MEMCLK 1b 1 MEMCLK
12:8	<b>CsOdtFineDelay: CS/ODT fine delay.</b> Read-write. Cold reset: 00h. This field specifies the time that the CS and ODT pins are delayed from the default setup time. 0_0000b No delay 0_0001b 1/64 MEMCLK delay 0_0010b 2/64 MEMCLK delay ... 1_1111b 31/64 MEMCLK delay
7:6	Reserved.
5	<b>CkeSetup: CKE setup time.</b> Read-write. Reset: 0. This bit selects the default setup time for the CKE pins versus MEMCLK. 0b 1/2 MEMCLK 1b 1 MEMCLK
4:0	<b>CkeFineDelay: CKE fine delay.</b> Read-write. Cold reset: 00h. This field specifies the time that the CKE pins are delayed from the default setup time. 0_0000b No delay 0_0001b 1/64 MEMCLK delay 0_0010b 2/64 MEMCLK delay ... 1_1111b 31/64 MEMCLK delay

### **F2x[1, 0]9C\_x[3:0]0[6:5] DRAM Read DQS Timing Control [High:Low] Registers**

Cold reset: 1F1F 1F1Fh. See `F2x[1, 0]98` for register access information.

DctOffset	Register
0000_0005h	DRAM Read DQS Timing Control Low DIMM 0: (bytes 0,1,2,3)
0000_0006h	DRAM Read DQS Timing Control High DIMM 0: (bytes 4,5,6,7)
0000_0007h	DRAM Read DQS ECC Timing Control DIMM 0
0000_0105h	DRAM Read DQS Timing Control Low DIMM 1: (bytes 0,1,2,3)
0000_0106h	DRAM Read DQS Timing Control High DIMM 1: (bytes 4,5,6,7)
0000_0107h	DRAM Read DQS ECC Timing Control DIMM 1
0000_0205h	DRAM Read DQS Timing Control Low DIMM 2: (bytes 0,1,2,3)
0000_0206h	DRAM Read DQS Timing Control High DIMM 2: (bytes 4,5,6,7)
0000_0207h	DRAM Read DQS ECC Timing Control DIMM 2
0000_0305h	DRAM Read DQS Timing Control Low DIMM 3: (bytes 0,1,2,3)
0000_0306h	DRAM Read DQS Timing Control High DIMM 3: (bytes 4,5,6,7)
0000_0307h	DRAM Read DQS ECC Timing Control DIMM 3

These registers control the nominal timing of read (input) DQS signals with respect to data. The actual delay applied to the DQS input signal before sampling data includes an internal part dependent delay plus the nominal register delay specified here. The part dependent (insertion) delay is large in proportion to individual step delay controlled by this register. See [2.8.9.9 \[DRAM Training\]](#) for information on how to use these registers. F2[1, 0]9C\_x[305, 205, 105, 05] are the DRAM Read DQS Timing Control Low Registers; they control DQS for bytes[3:0] of data. F2[1, 0]9C\_x[306, 206, 106, 06] are the DRAM Read DQS Timing Control High Registers; they control DQS for bytes[7:4] of data. The delay resolution is dependant upon the operating MEMCLK frequency. See [F2x\[1, 0\]94\[MemClkFreq\]](#). Each of the fields in these registers specify how much DQS is delayed nominally using delay hardware as follows:

- For memory clock frequencies less than 400 MHz, delay = RdDqsTimeByte \* 1/128 MEMCLKs, ranging from 0 to 63/128 MEMCLKs.
- For memory clock frequencies of 400 MHz or greater, delay = (RdDqsTimeByte & 03Eh) \* 1/128 MEMCLKs, ranging from 0 to 62/128 MEMCLKs ((writes to the LSB of RdDqsTimeByte are ignored and reads return zero).

To ensure unique values are written to each timing control register, BIOS must program these registers in consecutive DIMM order; i.e., program DIMM 0 register values first followed by the DIMM 1 values, etc.

Bits	Description
31:30	Reserved.
29:24	<b>RdDqsTimeByte[7, 3]: read DQS byte [7, 3] timing control.</b> Read-write.
23:22	Reserved.
21:16	<b>RdDqsTimeByte[6, 2]: read DQS byte [6, 2] timing control.</b> Read-write.
15:14	Reserved.
13:8	<b>RdDqsTimeByte[5, 1]: read DQS byte [5, 1] timing control.</b> Read-write.
7:6	Reserved.
5:0	<b>RdDqsTimeByte[4, 0]: read DQS byte [4, 0] timing control.</b> Read-write.

### **F2x[1, 0]9C\_x[3:0]07 DRAM Read DQS ECC Timing Control Register**

Cold reset: 0000 001Fh. See [F2x\[1, 0\]98](#) for register access information. See [F2x\[1, 0\]9C\\_x\[3:0\]0\[6:5\]](#) for a partially expanded list including DRAM Read DQS ECC Timing Registers and for revision specific information. See [2.8.9.9 \[DRAM Training\]](#) for information on how to use this register.

Bits	Description
31:6	Reserved.
5:0	<b>RdDqsTimeCheck: read DQS ECC byte timing control.</b> Read-write. This field specifies the delay that is added to the DQS signal associated with the ECC bits with respect to the data. The delay resolution is dependant upon the operating MEMCLK frequency, <a href="#">F2x[1, 0]94[MemClkFreq]</a> , as follows: <ul style="list-style-type: none"> <li>• For memory clock frequencies less than 400 MHz, delay = RdDqsTimeCheck * 1/128 MEMCLKs, ranging from 0 to 63/128 MEMCLKs.</li> <li>• For memory clock frequencies of 400 MHz or greater, delay = (RdDqsTimeCheck &amp; 03Eh) * 1/128 MEMCLKs, ranging from 0 to 62/128 MEMCLKs (writes to the LSB of RdDqsTimeByte are ignored and reads return zero).</li> </ul>

### F2x[1, 0]9C\_x08 DRAM Phy Control Register

Cold reset: 0208 0000h. See [F2x\[1, 0\]98](#) for register access information. See [2.8.9.9 \[DRAM Training\]](#) for information on how to use this register.

Bits	Description
31	Reserved.
30	<b>DisAutoComp: disable automatic compensation.</b> Read-write. 1=Disable the compensation control state machine. 0=The phy automatic compensation engine is enabled. Setting this bit in DCT0 disables the compensation engine for DCT0 and DCT1. Setting this bit in DCT1 has no effect.
29:14	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. The Phy stops the phase recovery engine during DQS receiver training. This allows the BIOS to reliably read the DQS receiver training data.
12	<b>WrLvOdtEn: write levelization ODT enabled.</b> Read-write. 1=ODT enabled during write levelization training. 0=No ODT is used for write levelization training.
11:8	<b>WrLvOdt[3:0]: write levelization ODT.</b> Read-write. This field specifies the state of the ODT pins that are driven out when WrLvOdtEn is set. For each bit, 1=ODT is enabled; 0=ODT is disabled. Tri-state enable for ODT is turned off by the phy while WrLvOdtEn is set. See <a href="#">2.8.9.5.7 [DRAM ODT Control]</a> .
7:6	Reserved.
5:4	<b>TrDimmSel: training DIMM select.</b> Read-write. This specifies which DIMM is to be trained. 00b=DIMM 0. 01b=DIMM 1. 10b=DIMM 2. 11b=DIMM 3. DIMM numbers are specified by <a href="#">[The DRAM CS Base Address Registers] F2x[1, 0][5C:40]</a> . For DDR3, bit[5] is reserved.
3	<b>PhyFenceTrEn: phy fence training enable.</b> Revision B: Write-only. Revision C and D: 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. This specifies nibbles of each DIMM data and ECC byte trained during write levelization training. 0=Lower nibbles. 1=Upper nibbles.
1	<b>WrtLvTrMode: write levelization training mode.</b> Read-write. 1=Write levelization training is done by the BIOS. 0=Write training is done by the hardware.
0	<b>WrtLvTrEn: write levelization training enable.</b> Read-write. 1=Initiate write levelization (tDQSS margining) training. 0=Stop write levelization training. The Phy stops the phase recovery engine during write levelization training. This allows the BIOS to reliably read the write levelization training data.



### F2x9C\_x09 DRAM Phy Driver Calibration Register

Cold reset: xxxx xxxh. See F2x[1, 0]98 for register access information. This register and F2x9C\_x0A are used by BIOS to program the phy's pre-driver calibration codes based on non-linear driver calibration codes read from this register. See 2.8.9.3 [Phy compensation initialization] for more information on how to program these registers.

Note: BIOS must not write to this register.

Bits	Description
31:30	Reserved.
29:25	<b>D3Cmp2DrvPCal: D3CMP 2 driver PMOS calibration code.</b> Read-write.
24:20	<b>D3Cmp2DrvNCal: D3CMP 2 driver NMOS calibration code.</b> Read-write.
19:15	<b>D3Cmp1DrvPCal: D3CMP 1 driver PMOS calibration code.</b> Read-write.
14:10	<b>D3Cmp1DrvNCal: D3CMP 1 driver NMOS calibration code.</b> Read-write.
9:5	<b>D3Cmp0DrvPCal: D3CMP 0 driver PMOS calibration code.</b> Read-write.
4:0	<b>D3Cmp0DrvNCal: D3CMP 0 driver NMOS calibration code.</b> Read-write.

### F2x9C\_x0A DRAM Phy Predriver Calibration Register

Cold reset: 0631 8C63h. See F2x[1, 0]98 for register access information. See 2.8.9.3 [Phy compensation initialization] for information on how to program this register.

Bits	Description										
31:28	Reserved.										
27:25	<p><b>D3Cmp2PCal: D3CMP 2 PMOS predriver calibration code.</b> Read-write. This field affects slew rates for all data and dqs pins associated with byte lanes 4 thru 7 and clock pads MEMCLK0_H/L[7:2] and MEMCLK1_H/L[7:2]. See 2.8.2 [DDR Pad to Processor Pin Mapping]. The recommended calibration code value programmed into this field corresponds to the normalized drive strength value programmed in F2x[1, 0]9C_x00[DataDrvStren] as specified below:</p> <table border="1"> <thead> <tr> <th><u>DataDrvStren</u></th> <th><u>Calibration Code</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>7h</td> </tr> <tr> <td>01b</td> <td>See note 1.</td> </tr> <tr> <td>10b</td> <td>5h</td> </tr> <tr> <td>11b</td> <td>3h</td> </tr> </tbody> </table> <p>1. If G34 processor populated with 3 DIMMS per channel at 1.5V, then Code=5h; elseif C32 processor populated with 3 SR DDR-1066 DIMMs per channel at 1.5V, then Code=1h; else Code=7h.</p>	<u>DataDrvStren</u>	<u>Calibration Code</u>	00b	7h	01b	See note 1.	10b	5h	11b	3h
<u>DataDrvStren</u>	<u>Calibration Code</u>										
00b	7h										
01b	See note 1.										
10b	5h										
11b	3h										
24:23	Reserved.										

22:20	<p><b>D3Cmp2NCal: D3CMP 2 NMOS predriver calibration code.</b> Read-write. This field affects slew rates for all data and dqs pins associated with byte lanes 4 thru 7 and clock pads MEMCLK0_H/L[7:2] and MEMCLK1_H/L[7:2]. See 2.8.2 [DDR Pad to Processor Pin Mapping]. The recommended calibration code value programmed into this field corresponds to the normalized drive strength value programmed in F2x[1, 0]9C_x00[DataDrvStren] as specified below:</p> <table border="1"> <thead> <tr> <th><u>DataDrvStren</u></th> <th><u>Calibration Code</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>7h</td> </tr> <tr> <td>01b</td> <td>See note 1.</td> </tr> <tr> <td>10b</td> <td>3h</td> </tr> <tr> <td>11b</td> <td>2h</td> </tr> </tbody> </table> <p>1. If G34 processor populated with 3 DIMMS per channel at 1.5V, then Code=5h; elseif C32 processor populated with 3 SR DDR-1066 DIMMs per channel at 1.5V, then Code=1h; else Code=7h.</p>	<u>DataDrvStren</u>	<u>Calibration Code</u>	00b	7h	01b	See note 1.	10b	3h	11b	2h
<u>DataDrvStren</u>	<u>Calibration Code</u>										
00b	7h										
01b	See note 1.										
10b	3h										
11b	2h										
19:18	Reserved.										
17:15	<p><b>D3Cmp1PCal: D3CMP 1 PMOS predriver calibration code.</b> Read-write. This field affects slew rates for command and address pins. The recommended calibration code value programmed into this field corresponds to the normalized drive strength value programmed in F2x[1, 0]9C_x00[AddrCmdDrvStren] as specified below:</p> <table border="1"> <thead> <tr> <th><u>AddrCmdDrvStren</u></th> <th><u>Calibration Code</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>7h</td> </tr> <tr> <td>01b</td> <td>5h</td> </tr> <tr> <td>10b</td> <td>3h</td> </tr> <tr> <td>11b</td> <td>2h</td> </tr> </tbody> </table>	<u>AddrCmdDrvStren</u>	<u>Calibration Code</u>	00b	7h	01b	5h	10b	3h	11b	2h
<u>AddrCmdDrvStren</u>	<u>Calibration Code</u>										
00b	7h										
01b	5h										
10b	3h										
11b	2h										
14:13	Reserved.										
12:10	<p><b>D3Cmp1NCal: D3CMP 1 NMOS predriver calibration code.</b> Read-write. This field affects slew rates for command and address pins. The recommended calibration code value programmed into this field corresponds to the normalized drive strength value programmed in F2x[1, 0]9C_x00[AddrCmdDrvStren] as specified below:</p> <table border="1"> <thead> <tr> <th><u>AddrCmdDrvStren</u></th> <th><u>Calibration Code</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>7h</td> </tr> <tr> <td>01b</td> <td>3h</td> </tr> <tr> <td>10b</td> <td>2h</td> </tr> <tr> <td>11b</td> <td>2h</td> </tr> </tbody> </table>	<u>AddrCmdDrvStren</u>	<u>Calibration Code</u>	00b	7h	01b	3h	10b	2h	11b	2h
<u>AddrCmdDrvStren</u>	<u>Calibration Code</u>										
00b	7h										
01b	3h										
10b	2h										
11b	2h										
9:8	Reserved.										
7:5	<p><b>D3Cmp0PCal: D3CMP 0 PMOS predriver calibration code.</b> Read-write. This field affects slew rates for all data and dqs pins associated with byte lanes 0 thru 3, the ECC byte lane, and clock pads MEMCLK0_H/L[1:0] and MEMCLK1_H/L[1:0]. See 2.8.2 [DDR Pad to Processor Pin Mapping]. The recommended calibration code value programmed into this field corresponds to the normalized drive strength value programmed in F2x[1, 0]9C_x00[DataDrvStren] as specified below:</p> <table border="1"> <thead> <tr> <th><u>DataDrvStren</u></th> <th><u>Calibration Code</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>7h</td> </tr> <tr> <td>01b</td> <td>See note 1.</td> </tr> <tr> <td>10b</td> <td>5h</td> </tr> <tr> <td>11b</td> <td>3h</td> </tr> </tbody> </table> <p>1. If G34 processor populated with 3 DIMMS per channel at 1.5V, then Code=5h; elseif C32 processor populated with 3 SR DDR-1066 DIMMs per channel at 1.5V, then Code=1h; else Code=7h.</p>	<u>DataDrvStren</u>	<u>Calibration Code</u>	00b	7h	01b	See note 1.	10b	5h	11b	3h
<u>DataDrvStren</u>	<u>Calibration Code</u>										
00b	7h										
01b	See note 1.										
10b	5h										
11b	3h										

4:3	Reserved.										
2:0	<p><b>D3Cmp0NCal: D3CMP 0 NMOS predriver calibration code.</b> Read-write. This field affects slew rates for all data and dqs pins associated with byte lanes 0 thru 3, the ECC byte lane, and clock pads MEMCLK0_H/L[1:0] and MEMCLK1_H/L[1:0]. See <a href="#">2.8.2 [DDR Pad to Processor Pin Mapping]</a>. The recommended calibration code value programmed into this field corresponds to the normalized drive strength value programmed in F2x[1, 0]9C_x00[DataDrvStren] as specified below:</p> <table border="1"> <thead> <tr> <th><u>DataDrvStren</u></th> <th><u>Calibration Code</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>7h</td> </tr> <tr> <td>01b</td> <td>See note 1.</td> </tr> <tr> <td>10b</td> <td>3h</td> </tr> <tr> <td>11b</td> <td>2h</td> </tr> </tbody> </table> <p>1. If G34 processor populated with 3 DIMMS per channel at 1.5V, then Code=5h; elseif C32 processor populated with 3 SR DDR-1066 DIMMs per channel at 1.5V, then Code=1h; else Code=7h.</p>	<u>DataDrvStren</u>	<u>Calibration Code</u>	00b	7h	01b	See note 1.	10b	3h	11b	2h
<u>DataDrvStren</u>	<u>Calibration Code</u>										
00b	7h										
01b	See note 1.										
10b	3h										
11b	2h										

### **F2x[1, 0]9C\_x0C DRAM Phy Miscellaneous Register**

Cold reset: 0013 0000h. See [F2x\[1, 0\]98](#) for register access information. This register provides access to the DDR phy to control signal tri-state functionality. See [Table 84](#) for processor pin map. Based on the system configuration, BIOS may tri-state signals with associated chip selects that are unpopulated in an effort to conserve power. The recommendations for tri-state of ODT pins are as follows:

- BIOS tri-states ODT pins to unpopulated DIMM slots.
- BIOS tri-states ODT pins that are unused on slots with unbuffered DIMMs.
- BIOS does not tri-state any ODT pins to slots with registered DIMMs.

This register also provides access to the DDR phy fence logic used to adjust the phase relationship between the data FIFO and the data going to the pad. See [2.8.9.7 \[Phy Fence programming\]](#) for information on how to program this register.

Software must rewrite the desired value to [F2x\[1,0\]9C\\_x0D0F0\[F:0\]30\[PwrDn\]](#) after any write to this register.

Bits	Description						
31:21	Reserved.						
20:16	<p><b>PhyFence: phy fence.</b> Read-write. This field specifies the fence delay value between the phy data FIFO, and the DDR pads. Fence delay = PhyFence * 1/64 of a MEMCLK, ranging from 0/64 to 31/64 MEMCLKs.</p>						
15:14	Reserved.						
13:12	<p><b>CKETri: CKE tri-state.</b> Read-write. 0=The CKE signals are not tri-stated unless directed to by the DCT. 1=Tri-state unconnected CKE signals from the processor. The bits of this field map to pads as follows:</p> <table border="1"> <thead> <tr> <th><u>Bit</u></th> <th><u>Package pin name</u></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>MEMCKE[1,0][0]</td> </tr> <tr> <td>1</td> <td>MEMCKE[1,0][1]</td> </tr> </tbody> </table> <p>See section <a href="#">2.8.2 [DDR Pad to Processor Pin Mapping]</a>.</p>	<u>Bit</u>	<u>Package pin name</u>	0	MEMCKE[1,0][0]	1	MEMCKE[1,0][1]
<u>Bit</u>	<u>Package pin name</u>						
0	MEMCKE[1,0][0]						
1	MEMCKE[1,0][1]						

11:8	<p><b>ODTTri: ODT tristate.</b> Read-write. 0=The ODT signals are not tristated unless directed to by the DCT. 1=Tri-state unconnected ODT signals from the processor. The bits of this field map to pads as follows:</p> <table border="0"> <tr> <td><u>Bit</u></td> <td><u>Pad</u></td> <td><u>Bit</u></td> <td><u>Pad</u></td> </tr> <tr> <td>0</td> <td>MEMODT[1,0][0]</td> <td>2</td> <td>MEMODT[1,0][2]</td> </tr> <tr> <td>1</td> <td>MEMODT[1,0][1]</td> <td>3</td> <td>MEMODT[1,0][3]</td> </tr> </table> <p>See section 2.8.2 [DDR Pad to Processor Pin Mapping].</p>	<u>Bit</u>	<u>Pad</u>	<u>Bit</u>	<u>Pad</u>	0	MEMODT[1,0][0]	2	MEMODT[1,0][2]	1	MEMODT[1,0][1]	3	MEMODT[1,0][3]								
<u>Bit</u>	<u>Pad</u>	<u>Bit</u>	<u>Pad</u>																		
0	MEMODT[1,0][0]	2	MEMODT[1,0][2]																		
1	MEMODT[1,0][1]	3	MEMODT[1,0][3]																		
7:0	<p><b>ChipSelTri: chip select tristate.</b> Read-write. 0=The chip select signals are not tristated unless directed to by the DCT. 1=Tristate unpopulated chip selects when motherboard termination is available. For single rank registered DIMMs with address parity capability, BIOS must not tri-state the chip select pin corresponding to the second chip select of the DIMM . The bit of this field map to pads as follows:</p> <table border="0"> <tr> <td><u>Bit</u></td> <td><u>Pad</u></td> <td><u>Bit</u></td> <td><u>Pad</u></td> </tr> <tr> <td>0</td> <td>MEMCS[1,0]_L[0]</td> <td>4</td> <td>MEMCS[1,0]_L[4]</td> </tr> <tr> <td>1</td> <td>MEMCS[1,0]_L[1]</td> <td>5</td> <td>MEMCS[1,0]_L[5]</td> </tr> <tr> <td>2</td> <td>MEMCS[1,0]_L[2]</td> <td>6</td> <td>MEMCS[1,0]_L[6]</td> </tr> <tr> <td>3</td> <td>MEMCS[1,0]_L[3]</td> <td>7</td> <td>MEMCS[1,0]_L[7]</td> </tr> </table> <p>See section 2.8.2 [DDR Pad to Processor Pin Mapping].</p>	<u>Bit</u>	<u>Pad</u>	<u>Bit</u>	<u>Pad</u>	0	MEMCS[1,0]_L[0]	4	MEMCS[1,0]_L[4]	1	MEMCS[1,0]_L[1]	5	MEMCS[1,0]_L[5]	2	MEMCS[1,0]_L[2]	6	MEMCS[1,0]_L[6]	3	MEMCS[1,0]_L[3]	7	MEMCS[1,0]_L[7]
<u>Bit</u>	<u>Pad</u>	<u>Bit</u>	<u>Pad</u>																		
0	MEMCS[1,0]_L[0]	4	MEMCS[1,0]_L[4]																		
1	MEMCS[1,0]_L[1]	5	MEMCS[1,0]_L[5]																		
2	MEMCS[1,0]_L[2]	6	MEMCS[1,0]_L[6]																		
3	MEMCS[1,0]_L[3]	7	MEMCS[1,0]_L[7]																		

### F2x[1, 0]9C\_x0D DRAM Phy DLL Control Register

Revision B: Reserved. Revision C and later: Cold reset: 0000 0000h. See F2x[1, 0]98 for register access information. 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 to its cold reset value during usage of the PRE (see F2x[1, 0]9C\_x[51:50]). See 2.8.9.6 [DRAM Device and Controller Initialization] for more information.

Bits	Description
31:26	Reserved.
25:24	<b>RxDLLWakeupTime: receive DLL wakeup time.</b> Read-write. This field specifies the number of PCLKs that the DLL standby signal must deassert prior to a DLL relock event or before read traffic is sent to the receive DLLs.
23	Reserved.
22:20	<b>RxCPUpdPeriod: receive charge pump period.</b> Read-write. This field specifies the number of DLL relocks required to keep the receive DLLs locked for the period where there is no read traffic.
19	Reserved.
18:16	<b>RxMaxDurDllNoLock: receive maximum duration DLL no lock.</b> Read-write. This field 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. BIOS should leave this field at the default value of 0.
15:10	Reserved.
9:8	<b>TxDLLWakeupTime: transmit DLL wakeup time.</b> Read-write. This field specifies the number of PCLK's that the DLL standby signal must deassert prior to a DLL relock event or before write traffic is sent to transmit DLLs.
7	Reserved.

6:4	<b>TxCPUpdPeriod: transmit charge pump DLL wakeup time.</b> Read-write. This specifies the number of DLL relocks required to keep the TxDLLs locked for the period where there is no write traffic.
3	Reserved.
2:0	<b>TxMaxDurDllNoLock: transmit maximum duration DLL no lock.</b> Read-write. This field 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. BIOS should leave this field at the default value of 0.

### **F2x[1, 0]9C\_x[2B:10] DRAM DQS Receiver Enable Timing Control Registers**

See [F2x\[1, 0\]98](#) for register access information.

These registers are organized as eight groups of registers, two groups for each DIMM on a channel. DIMM numbers are specified by [\[The DRAM CS Base Address Registers\] F2x\[1, 0\]\[5C:40\]](#). The definition of [F2x\[1, 0\]98\[DctOffset\]](#) registers 0000\_0010h through 0000\_001Bh varies with [F2x\[1, 0\]94\[LegacyBiosMode\]](#). To ensure unique values are written to each DQS receiver enable timing control register, BIOS must program these registers in consecutive order; i.e., program DIMM 0 register values first followed by the DIMM 1 values, etc.

<u>DctOffset</u>	<u>Register</u>
0000_0010h	DRAM DQS Receiver Enable Timing Control Low DIMM 0: (bytes 0,1)
0000_0011h	DRAM DQS Receiver Enable Timing Control High DIMM 0: (bytes 2,3)
0000_0012h	DRAM DQS Receiver Enable Timing Control ECC DIMM 0
0000_0013h	DRAM DQS Receiver Enable Timing Control Low DIMM 1: (bytes 0,1)
0000_0014h	DRAM DQS Receiver Enable Timing Control High DIMM 1: (bytes 2,3)
0000_0015h	DRAM DQS Receiver Enable Timing Control ECC DIMM 1
0000_0016h	DRAM DQS Receiver Enable Timing Control Low DIMM 2: (bytes 0,1)
0000_0017h	DRAM DQS Receiver Enable Timing Control High DIMM 2: (bytes 2,3)
0000_0018h	DRAM DQS Receiver Enable Timing Control ECC DIMM 2
0000_0019h	DRAM DQS Receiver Enable Timing Control Low DIMM 3: (bytes 0,1)
0000_001Ah	DRAM DQS Receiver Enable Timing Control High DIMM 3: (bytes 2,3)
0000_001Bh	DRAM DQS Receiver Enable Timing Control ECC DIMM 3
0000_0020h	DRAM DQS Receiver Enable Timing Control Low DIMM 0: (bytes 4,5)
0000_0021h	DRAM DQS Receiver Enable Timing Control High DIMM 0: (bytes 6,7)
0000_0022h	Reserved
0000_0023h	DRAM DQS Receiver Enable Timing Control Low DIMM 1: (bytes 4,5)
0000_0024h	DRAM DQS Receiver Enable Timing Control High DIMM 1: (bytes 6,7)
0000_0025h	Reserved
0000_0026h	DRAM DQS Receiver Enable Timing Control Low DIMM 2: (bytes 4,5)
0000_0027h	DRAM DQS Receiver Enable Timing Control High DIMM 2: (bytes 6,7)
0000_0028h	Reserved
0000_0029h	DRAM DQS Receiver Enable Timing Control Low DIMM 3: (bytes 4,5)
0000_002Ah	DRAM DQS Receiver Enable Timing Control High DIMM 3: (bytes 6,7)
0000_002Bh	Reserved

Each of these registers control the timing of the receiver enable with respect to MEMCLK. The delay is applied from the falling MEMCLK edge following the CAS latency minus 1 rising edge of a read command. See [2.8.9.9 \[DRAM Training\]](#) for information on how to use these registers. Individual controls for each byte of data and ECC data are provided. Each control includes a gross timing field and a fine timing field, the sum of

which is the total delay. They are defined as follows:

Fine timing (for DqsRcvEnFineDelay and DqsRcvEnFineDelayCheck):

Delay = DqsRcvEnFineDelay \* 1/64 MEMCLKs, ranging from 0 to 31/64 MEMCLKs.

Gross timing:

0000b	No delay	1000b	4.0 MEMCLK delay
0001b	0.5 MEMCLK delay	1001b	4.5 MEMCLK delay
0010b	1.0 MEMCLK delay	1010b	5.0 MEMCLK delay
0011b	1.5 MEMCLK delay	1011b	5.5 MEMCLK delay
0100b	2.0 MEMCLK delay	1100b	6.0 MEMCLK delay
0101b	2.5 MEMCLK delay	1101b	6.5 MEMCLK delay
0110b	3.0 MEMCLK delay	1110b	7.0 MEMCLK delay
0111b	3.5 MEMCLK delay	1111b	7.5 MEMCLK delay
1xxx b	Reserved on revision B and C processors.		

**DRAM DQS Receiver Enable Timing Control [High, Low] Registers** (For DctOffset registers 0000\_0010h through 0000\_001Bh, the high registers apply to bytes[3:2] and the low registers apply to bytes[1:0]. For DctOffset registers 0000\_0020h through 0000\_002Bh, the high registers apply to bytes[7:6] and the low registers apply to bytes[5:4].)

Bits	Description
31:29	Reserved.
28:25	Reserved.
24:21	<b>DqsRcvEnGrossDelay[7, 5],[3, 1]: DQS receiver enable gross delay[7, 5],[3, 1].</b> Read-write. Reset: 0.
20:16	<b>DqsRcvEnFineDelay[7, 5],[3, 1]: DQS receiver enable fine delay[7, 5],[3, 1].</b> Read-write. Cold reset: 0.
15:13	Reserved.
12:9	Reserved.
8:5	<b>DqsRcvEnGrossDelay[6, 4],[2, 0]: DQS receiver enable gross delay[6, 4],[2, 0].</b> Read-write. Reset: 0.
4:0	<b>DqsRcvEnFineDelay[6, 4],[2, 0]: DQS receiver enable fine delay[6, 4],[2, 0].</b> Read-write. Cold reset: 0.

#### DRAM DQS Receiver Enable Timing Control ECC Registers

See [F2x\[1, 0\]98](#) for register access information.

Bits	Description
31:9	Reserved.
8:5	<b>DqsRcvEnGrossDelayCheck: DQS receiver enable gross delay ECC.</b> Read-write. Reset: 0.
4:0	<b>DqsRcvEnFineDelayCheck: DQS receiver enable fine delay ECC.</b> Read-write. Cold reset: 0.

#### **F2x[1, 0]9C\_x[4A:30] DRAM DQS Write Timing Control Registers**

These registers are valid only when [F2x\[1, 0\]94\[Ddr3Mode\] = 1](#) and should be programmed to 0 otherwise. There are four groups of registers for each DDR3 DIMM. DIMM numbers are specified by [\[The DRAM CS Base Address Registers\] F2x\[1, 0\]\[5C:40\]](#).



<u>DctOffset</u>	<u>Register</u>
0000_0030h	DRAM DQS Write Timing Control Low DIMM 0: (bytes 0,1)
0000_0031h	DRAM DQS Write Timing Control High DIMM 0: (bytes 2,3)
0000_0032h	DRAM DQS Write Timing Control ECC DIMM 0
0000_0033h	DRAM DQS Write Timing Control Low DIMM 1: (bytes 0,1)
0000_0034h	DRAM DQS Write Timing Control High DIMM 1: (bytes 2,3)
0000_0035h	DRAM DQS Write Timing Control ECC DIMM 1
0000_0036h	DRAM DQS Write Timing Control Low DIMM 2: (bytes 0,1)
0000_0037h	DRAM DQS Write Timing Control High DIMM 2: (bytes 2,3)
0000_0038h	DRAM DQS Write Timing Control ECC DIMM 2
0000_0039h	DRAM DQS Write Timing Control Low DIMM 3: (bytes 0,1)
0000_003Ah	DRAM DQS Write Timing Control High DIMM 3: (bytes 2,3)
0000_003Bh	DRAM DQS Write Timing Control ECC DIMM 3
0000_0040h	DRAM DQS Write Timing Control Low DIMM 0: (bytes 4,5)
0000_0041h	DRAM DQS Write Timing Control High DIMM 0: (bytes 6,7)
0000_0042h	Reserved
0000_0043h	DRAM DQS Write Timing Control Low DIMM 1: (bytes 4,5)
0000_0044h	DRAM DQS Write Timing Control High DIMM 1: (bytes 6,7)
0000_0045h	Reserved
0000_0046h	DRAM DQS Write Timing Control Low DIMM 2: (bytes 4,5)
0000_0047h	DRAM DQS Write Timing Control High DIMM 2: (bytes 6,7)
0000_0048h	Reserved
0000_0049h	DRAM DQS Write Timing Control Low DIMM 3: (bytes 4,5)
0000_004Ah	DRAM DQS Write Timing Control High DIMM 3: (bytes 6,7)
0000_004Bh	Reserved

Each of these registers control the DQS timing delay for write commands relative to MEMCLK. See [2.8.9.9 \[DRAM Training\]](#) for information on how to use this register. To ensure unique values are written to each timing control register, BIOS must program these registers in consecutive DIMM order; i.e., program DIMM 0 register values first followed by the DIMM 1 values, etc.

Individual controls for each byte of data and ECC data are provided. Each control includes a gross timing field and a fine timing field, the sum of which is the total delay. They are defined as follows:

Fine timing (for WrDqsFineDly and WrDqsChkFineDly):

Delay = WrDqsFineDly \* 1/64 MEMCLKs, ranging from 0 to 31/64 MEMCLKs.

Delay = WrDqsFineDly \* 1/64 MEMCLKs, ranging from 0 to 31/64 MEMCLKs.

Gross timing:

000b	No delay
001b	0.5 MEMCLK delay
010b	1.0 MEMCLK delay
011b	1.5 MEMCLK delay
1xxb	Reserved

### DRAM DQS Write Timing Control [High, Low] Registers

See [F2x\[1, 0\]98](#) for register access information. (For DctOffset registers 0000\_0030h through 0000\_003Bh, the high registers apply to bytes[3:2] and the low registers apply to bytes[1:0]. For DctOffset registers 0000\_0040h through 0000\_004Bh, the high registers apply to bytes[7:6] and the low registers apply to bytes[5:4])



Bits	Description
31	Reserved.
30:29	Reserved. Read-write.
28:24	Reserved.
23:21	<b>WrDqsGrossDly[7, 5],[3, 1]: DQS write gross delay[7, 5],[3, 1].</b> Read-write. Reset: 0.
20:16	<b>WrDqsFineDly[7, 5],[3, 1]: DQS write fine delay[7, 5],[3, 1].</b> Read-write. Cold reset: 0.
15	Reserved.
14:13	Reserved.
12:8	Reserved.
7:5	<b>WrDqsGrossDly[6, 4],[2, 0]: DQS write gross delay[6, 4],[2, 0].</b> Read-write. Reset: 0.
4:0	<b>WrDqsFineDly[6, 4],[2, 0]: DQS write fine delay[6, 4],[2, 0].</b> Read-write. Cold reset: 0.

### DRAM DQS Write Timing Control ECC Registers

See [F2x\[1, 0\]98](#) for register access information.

Bits	Description
31:8	Reserved.
7:5	<b>WrDqsChkGrossDly: DQS write gross delay ECC.</b> Read-write. Reset: 0.
4:0	<b>WrDqsChkFineDly: DQS write fine delay ECC.</b> Read-write. Cold reset: 0.

### **F2x[1, 0]9C\_x[51:50] DRAM Phase Recovery Control Register [High:Low] Registers**

See [F2x\[1, 0\]98](#) for register access information. 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 [2.8.9.9 \[DRAM Training\]](#) for information on how to use these registers. F2[1, 0]9C\_x50 is the Low Register which controls bytes[3:0] of data. F2[1, 0]9C\_x51 is the High Register which controls bytes[7:4] of data. The fields in these registers are encoded as follows:

Gross timing (PhRecGrossDlyByte): indicates the number of half-MEMCLK periods that the phase recovery engine advanced while aligning edges, ranging from 0.0 to 1.5 MEMCLK periods.

Fine timing (PhRecFineDlyByte):

Delay = PhRecFineDlyByte \* 1/64 of a MEMCLK, ranging from 0/64 to 31/64 MEMCLKs.

The total delay is the sum of these two fields, ranging from 0 to 1 and 63/64 MEMCLKs.

Bits	Description
31	Reserved.
30:29	<b>PhRecGrossDlyByte[7, 3]: phase recovery gross delay byte [7, 3].</b> Read-write. Reset: X.
28:24	<b>PhRecFineDlyByte[7, 3]: phase recovery fine delay byte [7, 3].</b> Read-write. Reset: X.
23	Reserved.
22:21	<b>PhRecGrossDlyByte[6, 2]: phase recovery gross delay byte [6, 2].</b> Read-write. Reset: X.
20:16	<b>PhRecFineDlyByte[6, 2]: phase recovery fine delay byte [6, 2].</b> Read-write. Reset: X.

15	Reserved.
14:13	<b>PhRecGrossDlyByte[5, 1]: phase recovery gross delay byte [5, 1].</b> Read-write. Reset: X.
12:8	<b>PhRecFineDlyByte[5, 1]: phase recovery fine delay byte [5, 1].</b> Read-write. Reset: X.
7	Reserved.
6:5	<b>PhRecGrossDlyByte[4, 0]: phase recovery gross delay byte [4, 0].</b> Read-write. Reset: X.
4:0	<b>PhRecFineDlyByte[4, 0]: phase recovery fine delay byte [4, 0].</b> Read-write. Reset: X.

### **F2x[1, 0]9C\_x52 DRAM ECC Phase Recovery Control Register**

Reset: see field definitions. See [F2x\[1, 0\]98](#) for register access information. This register provides the same function as [F2x\[1, 0\]9C\\_x\[51:50\]](#) for the ECC bits of the interface; see that register for more information.

Bits	Description
31:7	Reserved.
6:5	<b>PhRecEccGrossDlyByte: phase recovery ECC gross delay byte.</b> Read-write. Reset: X.
4:0	<b>PhRecEccFineDlyByte: phase recovery ECC fine delay byte.</b> Read-write. Reset: X.

### **F2x[1, 0]9C\_x53 Write Levelization Error Register**

Reset: see field definitions. See [F2x\[1, 0\]98](#) for register access information. This register is used by BIOS for hardware assisted DRAM training. See [2.8.9.9 \[DRAM Training\]](#) for information on how to use this register.

Bits	Description
31:9	Reserved.
8:0	<b>WrtLvErr: write levelization error.</b> Read-only. Reset: X. This field indicates the phase recovery error state which is used by BIOS for write levelization training for each byte of data and ECC. Bit[0] applies to byte0; bit[1] applies to byte1; etc. Bit[8] indicates the ECC byte state.

### **F2x[1,0]9C\_x0D0F0[F,8:0]0[8,4,0] Data Byte Group [2:0] Pad Configuration**

Revision B and earlier, RB-C2, and BL-C2: Reserved. Revision DA-C, C3 and later: See [F2x\[1, 0\]98](#) for register access information. This register controls impedance for the DQ[7:0,DQS, DM] pins and should be used only to override the default settings. The value of [DctOffset\[11:8\]](#) specifies the register instance while [DctOffset\[3:0\]](#) specifies the group within each instance, as follows:

<a href="#">DctOffset[11:8]</a>	<a href="#">DctOffset[3:0]</a>
0h - 7h: Data byte lane	0h: Group 0, DQ[0,1,4,5] pins
8h: ECC byte lane	4h: Group 1, DQS and DM pins
Fh: all byte lanes	8h: Group 2, DQ[2,3,6,7] pins.

Bits	Description
15:3	Reserved.
2	<b>OdtModeInvert: ODT mode invert.</b> Read-write. Reset: 0. 1=Swap DDR2 and DDR3 encodings for <a href="#">F2x[1, 0]9C_x00[ProcODT]</a> . This allows DDR2 values to be used when <a href="#">F2x[1, 0]94[Ddr3Mode]=1</a> or DDR3 values to be used when <a href="#">F2x[1, 0]94[Ddr3Mode]=0</a> .
1:0	Reserved.

### **F2x[1,0]9C\_x0D0F0[F:0]13 Data Byte DLL Configuration Register**

Revision B, RB-C2, and BL-C2 and earlier: Reserved. Revision DA-C, C3 and later: See [F2x\[1, 0\]98](#) for register access information. The value of DctOffset[11:8] specifies the register instance accessed as follows:

<u>DctOffset[11:8]</u>	<u>Byte Lane</u>
0h - 7h	Data Byte 0 - Data Byte 7.
8h	ECC Byte.
Fh	Broadcast writes to all byte lanes. Not valid for reads.

All byte lanes should be programmed to the same value.

Bits	Description
15:8	Reserved.
7	<b>RxDqsUDllPowerDown: Rx Dqs upper DLL power down.</b> Read-write. Reset: 0. 1=Power down the upper receiver DQS DLL. See <a href="#">2.8.9.10 [DRAM Phy Power Savings]</a> .
6:0	Reserved.

### **F2x[1,0]9C\_x0D0F0[F:0]30 Data Byte Configuration Register**

Revision B, RB-C2, and BL-C2 and earlier: Reserved. Revision DA-C, C3 and later: See [F2x\[1, 0\]98](#) for register access information. The value of DctOffset[11:8] specifies the register instance accessed as follows:

<u>DctOffset[11:8]</u>	<u>Byte Lane</u>
0h - 7h	Data Byte 0 - Data Byte 7
8h	ECC Byte
Fh	Broadcast writes to all byte lanes. Not valid for reads.

Bits	Description
15:5	Reserved.
4	<b>PwrDn: power down.</b> Read-write. Reset: 0. 1=Turn off DLL circuitry for ECC bits. See <a href="#">2.8.9.10 [DRAM Phy Power Savings]</a> and <a href="#">F2x[1, 0]9C_x0C</a> .
3:0	Reserved.

### **F2x[1,0]9C\_x0D0F2[F:0]30 Clock Configuration**

Revision B, RB-C2, and BL-C2 and earlier: Reserved. Revision DA-C, C3 and later: See [F2x\[1, 0\]98](#) for register access information. The value of DctOffset[11:8] specifies the register instance accessed. Each register instance maps to a pair of clock pads as follows:

<u>DctOffset[11:8]</u>	<u>DCT0 pad</u>	<u>DCT1 pad</u>
0h	MEMCLK0_H[1:0]	MEMCLK1_H[1:0]
1h	MEMCLK0_H[5:4]	MEMCLK1_H[5,2]
2h	MEMCLK0_H[3:2]	MEMCLK1_H[4:3]
3h	MEMCLK0_H[7:6]	MEMCLK1_H[7:6]
Fh <sup>1</sup>	MEMCLK0_H[7:0]	MEMCLK1_H[7:0]

1. Broadcast write. Not valid for reads.

See [2.8.2 \[DDR Pad to Processor Pin Mapping\]](#) for pad to pin mapping.

Bits	Description
15:5	Reserved.

4	<b>PwrDn: power down.</b> Read-write. Reset: 0. 1=Turn off DLL circuitry for unused clocks. See <a href="#">2.8.9.10 [DRAM Phy Power Savings]</a> .
3:0	Reserved.

### **F2x[1,0]9C\_x0D0F[9:8][F:0]0[8,4,0] C/A Pad Configuration**

Reset: 0003h. See [F2x\[1, 0\]98](#) for register access information. DctOffset[15:0] specifies the instance and group number accessed as follows:

<u>DctOffset[15:0]</u>	<u>Pins</u>
8000h	CS_L[7,5,3,1]
8004h	ODT[3:0]
8008h	CS_L[6,4,2,0]
8100h	RAS_L, CAS_L, WE_L, and ADD[13]
8104h	BANK[1:0], ADD[10,0]
8108h	PAR

Bits	Description
15:14	Reserved.
13	<b>PowerDown.</b> Read-write. 1=Pad circuitry is turned off. 0=Pad circuitry is turned on. See <a href="#">2.8.9.10 [DRAM Phy Power Savings]</a> .
12:0	Reserved.

### **F2x[1,0]9C\_x0D0F812F C/A Tristate Control**

Revision B, RB-C2, and BL-C2: Reserved. Revision DA-C, C3 and later: Reset: 0000h. See [F2x\[1, 0\]98](#) for register access information

Bits	Description								
15:12	Reserved.								
11:0	<b>TristateCA[7:0]: Tri-state C/A [11:0].</b> Read-write. For each bit: 1=Pad associated with the pin is tri-stated for power savings. 0=Pad provides termination. BIOS should set this bit for unused pins. See <a href="#">2.8.9.10 [DRAM Phy Power Savings]</a> . <table> <thead> <tr> <th><u>Bit</u></th> <th><u>Pin</u></th> <th><u>Bit</u></th> <th><u>Pin</u></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>PAR</td> <td>1 -11</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bit</u>	<u>Pin</u>	<u>Bit</u>	<u>Pin</u>	0	PAR	1 -11	Reserved
<u>Bit</u>	<u>Pin</u>	<u>Bit</u>	<u>Pin</u>						
0	PAR	1 -11	Reserved						

### **F2x[1,0]9C\_x0D0FE006 Phy PLL Lock Time**

Revision B, RB-C2, and BL-C2 and earlier: Reserved. Revision DA-C, C3 and later: Cold reset: 0000 07D0h. See [F2x\[1, 0\]98](#) for register access information.

Bits	Description
15:0	<b>PLLLockTime: pll lock time.</b> Read-write. This field specifies the number of 200 MHz clock cycles the phy waits for PLLs to lock during a frequency change. See <a href="#">2.8.9.8 [DRAM Channel Frequency Change]</a> .

### **F2x[1,0]9C\_x0D0FE007 Phy DLL Lock Time**

Revision B, RB-C2, and BL-C2 and earlier: Reserved. Revision DA-C, C3 and later: Cold reset: 0190h. See

F2x[1, 0]98 for register access information.

Bits	Description
15:0	<b>DllLockTime: dll lock time.</b> Read-write. This field specifies the number of 200 MHz clock cycles the phy waits for DLLs to lock during a frequency change. See 2.8.9.8 [DRAM Channel Frequency Change].

### F2x[1, 0]A0 DRAM Controller Miscellaneous Register

Reset: 0000 0000h. See 2.8.1 [DCT Configuration Registers] for general programming information about DCT configuration registers.

Bits	Description
31:10	Reserved.
9	<b>DramEnabled: DRAM enabled.</b> Read-only. This bit is identical to F2x110[DramEnabled].
8:0	Reserved.

### F2x[1, 0]A4 DRAM Controller Temperature Throttle Register

Revisions C and earlier revisions: Reserved. Revision D and later: Reset: 0000 0000h. See 2.8.1 [DCT Configuration Registers] for general programming information about DCT configuration registers. See 2.8.13 [DRAM On DIMM Thermal Management and Power Capping] for programming information about this register.

Bits	Description
31:15	Reserved.
14:12	<b>CmdThrottleMode: command throttle mode.</b> Read-write. This specifies the command throttle mode when ODTSEn=1 and either the EVENT_L pin is asserted or BwCapEn=1. The DCT throttles commands over a rolling window of 100 clock cycles, maintaining the average throttling as specified by this field. 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 - Throttle commands by 100%. DRAM devices are placed into power down mode (see F2x[1, 0]94[PowerDownMode]). This mode should only be used to prevent physical damage as system timeouts may occur. This mode requires F2x[1, 0]94[PowerDownEn]=1 and F2x[1, 0]94[PowerDownMode]=0. This mode should not be used with BwCapEn. Note: Throttling should not be enabled until after DRAM initialization (F2x110[DramEnable]=1) and training (see 2.8.9.9 [DRAM Training]) are complete.
11	<b>BwCapEn: bandwidth capping enable.</b> Read-write. 1=The memory command throttle mode specified by CmdThrottleMode is applied. This bit is used by software to enable command throttling independent of the state of the EVENT_L pin.
10:9	Reserved.

8	<b>ODTSEn: on DIMM temperature sensor enable.</b> Read-write. This enables the monitoring of the EVENT_L pin and indicates whether the on DIMM temperature sensors of the DIMMs on a channel are 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]. 0 = ODTs disabled. 1 = ODTs enabled.
7:0	Reserved.

### F2x[1, 0]A8 DRAM Controller Miscellaneous Register 2

Reset: 0000 0000h. See 2.8.1 [DCT Configuration Registers] for general programming information about DCT configuration registers.

Bits	Description																								
31:19	Reserved.																								
18:16	<p><b>DataTxFifoWrDly: data transmit FIFO write delay.</b> Revision B: Reserved. Revision C and later: Read-write. This field causes the DRAM controller to delay sending write data to the phy to avoid FIFO overflow conditions.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>No delay.</td> </tr> <tr> <td>001b</td> <td>Delay by 0.5 MEMCLK.</td> </tr> <tr> <td>010b</td> <td>Delay by 1.0 MEMCLK.</td> </tr> <tr> <td>011b</td> <td>Delay by 1.5 MEMCLKs.</td> </tr> <tr> <td>100b</td> <td>Delay by 2.0 MEMCLKs.</td> </tr> </tbody> </table> <p>All other encodings are reserved. BIOS should program this field based on F2x[1, 0]78[RdPtrInit]:</p> <table border="1"> <thead> <tr> <th>RdPtrInit</th> <th>DataTxFifoWrDly</th> </tr> </thead> <tbody> <tr> <td>0110b</td> <td>000b</td> </tr> <tr> <td>0101b</td> <td>001b</td> </tr> <tr> <td>0100b</td> <td>If (F3x1F0[NbPstate] != 000b) then 001b; else 010b.</td> </tr> <tr> <td>0011b</td> <td>011b</td> </tr> <tr> <td>0010b</td> <td>100b</td> </tr> </tbody> </table>	Bits	Definition	000b	No delay.	001b	Delay by 0.5 MEMCLK.	010b	Delay by 1.0 MEMCLK.	011b	Delay by 1.5 MEMCLKs.	100b	Delay by 2.0 MEMCLKs.	RdPtrInit	DataTxFifoWrDly	0110b	000b	0101b	001b	0100b	If (F3x1F0[NbPstate] != 000b) then 001b; else 010b.	0011b	011b	0010b	100b
Bits	Definition																								
000b	No delay.																								
001b	Delay by 0.5 MEMCLK.																								
010b	Delay by 1.0 MEMCLK.																								
011b	Delay by 1.5 MEMCLKs.																								
100b	Delay by 2.0 MEMCLKs.																								
RdPtrInit	DataTxFifoWrDly																								
0110b	000b																								
0101b	001b																								
0100b	If (F3x1F0[NbPstate] != 000b) then 001b; else 010b.																								
0011b	011b																								
0010b	100b																								
15:8	<p><b>CtrlWordCS[7:0]: control word chip select.</b> Read-write. This field specifies the target DIMM chip selects used for control word programming. This field is used in conjunction with F2x[1, 0]7C[SendControlWord].</p> <p>00000011b - CS0,CS1 is asserted. 00001100b - CS2,CS3 is asserted. 00110000b - CS4,CS5 is asserted. 11000000b - CS6,CS7 is asserted. All other values are reserved.</p>																								
7	<p><b>ProgOdtEn: programmable ODT pattern enable.</b> Revision C and earlier revisions: Reserved. Revision D and later: Read-write. 1=The state of the ODT[3:0] pins is specified by F2x[1, 0]F4_x[181:180] and F2x[1, 0]F4_x[183:182]. F2x[1, 0]A8[OdtSwizzle] must be 0 when ProgOdtEn=1. See 2.8.9.5.7 [DRAM ODT Control]. BIOS should program this bit to 1 for DDR3 mode and 0 for DDR2 mode.</p>																								

6	<p><b>OdtSwizzle: ODT swizzle.</b> Revision B: Reserved. Revision C and later: Read-write. 1= The functionality of the second and third logical ODT pins is switched. On packages that support registered DIMMs, these are labeled as M[B, A]1_ODT[0] and M[B, A]2_ODT[0]. On packages that support unbuffered DIMMs these are labeled as M[B, A]1_ODT[0] and M[B, A]0_ODT[1]. 0 = The functionality of the ODT pins is not switched. BIOS is recommended to program this bit on platforms in accordance with the appropriate motherboard design guidelines as follows:</p> <ul style="list-style-type: none"> <li>• For DDR2, program OdtSwizzle=1 if ProgOdtEn=0 and F2x[1, 0]94[FourRankRDimm]=0 and F2x[1, 0]94[PowerDownEn]=1 and F2x[1, 0]94[PowerDownMode]=1.</li> <li>• For DDR3, program OdtSwizzle=1 if ProgOdtEn=0 and Ddr3FourSocketCh=0 and F2x[1, 0]94[FourRankRDimm]=0.</li> <li>• Program OdtSwizzle=0 for all other cases.</li> </ul>
5	<p><b>SubMemclkRegDly: Sub-one MEMCLK register delay.</b> Revision B: Reserved. Revision C and later revisions: Read-write. 1= The delay through the DIMM register and routing delay is less than 1 MEMCLK. The DCT does not add 1 MEMCLK to calculate write latency for DDR3 registered DIMMs. 0 = The delay through the DIMM register and routing delay is at least 1 MEMCLK. The DCT adds 1 MEMCLK to calculate write latency for DDR3 registered DIMMs.</p>
4:3	Reserved.
2	<p><b>Ddr3FourSocketCh: DDR3 four DIMMs per channel support.</b> Revision B: Reserved. Revision C and later: Read-write. 1= 3 or 4 DDR3 DIMM sockets are supported per channel. 0= Up to 2 DDR3 DIMM sockets are supported per channel. This bit must be programmed to the same state for both DRAM controllers.</p>
1	<p><b>ActiveCmdAtRst: active command signals reset.</b> Read-write. 1=Drive DDR command signals (RAS_L, CAS_L, WE_L) low when MEMRST_L is transitioning from low to high. Before CKE is asserted, drive the DDR command signals high. 0=Drive all DDR interface signals high when MEMRST_L is transitioning from low to high. BIOS should program this bit to 1 if F2x[1, 0]94[Ddr3Mode]=0 and F2x[1, 0]90[ParEn]=1; otherwise leave this bit at 0.</p>
0	Reserved.

### F2xAC DRAM Controller Temperature Status Register

Revision C and earlier revisions: Reserved. Revision D: Cold reset: 0000 0000h.

Bits	Description
31:3	Reserved.
2	<p><b>MemTempHot1: Memory temperature hot, DCT1.</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 F2x[1, 0]A4).</p>
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 F2x[1, 0]A4).</p>

### F2x[1, 0]F0 DRAM Controller Extra Data Offset Register

Revision C and earlier revisions: Reserved. Revision D and later: Reset: 8000 0000h. The DCTs each include an array of registers called F2x[1, 0]F4\_x[FFFFFFF:0], which are defined following F2x[1, 0]F4. [The DRAM Controller Extra Data Offset Register] F2x[1, 0]F0 and [The DRAM Controller Extra Data Port] F2x[1, 0]F4 are used to access F2x[1, 0]F4\_x[FFFFFFF:0]. The register number (i.e., the number that follows “\_x” in the



register mnemonic) is specified by  $F2x[1, 0]F0[DctOffset]$ . Access to these registers is accomplished as follows:

- Reads:
  - Write the register number to  $F2x[1, 0]F0[DctOffset]$  with  $F2x[1, 0]F0[DctAccessWrite]=0$ .
  - Poll  $F2x[1, 0]F0[DctAccessDone]$  until it is high.
  - Read the register contents from  $F2x[1, 0]F4$ .
- Writes:
  - Write all 32 bits to the register data to  $F2x[1, 0]F4$  (individual byte writes are not supported).
  - Write the register number to  $F2x[1, 0]F0[DctOffset]$  with  $F2x[1, 0]F0[DctAccessWrite]=1$ .
  - Poll  $F2x[1, 0]F0[DctAccessDone]$  until it is high to ensure that the contents of the write have been delivered.

Bits	Description
31	<b>DctAccessDone: DRAM controller access done.</b> Read-only. 1=The access to one of the $F2x[1, 0]F4\_x[FFFFFFF:0]$ registers is complete. 0=The access is still in progress.
30	<b>DctAccessWrite: DRAM controller read/write select.</b> Read-write. 0=Read one of the $F2x[1, 0]F4\_x[FFFFFFF:0]$ registers. 1=Write one of the $F2x[1, 0]F4\_x[FFFFFFF:0]$ registers.
29:28	Reserved.
27:0	<b>DctOffset: DRAM controller offset.</b> Read-write.

### **$F2x[1, 0]F4$ DRAM Controller Extra Data Port**

See  $F2x[1, 0]F0$  for details about this port.

### **$F2x[1, 0]F4\_x[181:180]$ DCT ODT Control [High:Low] Registers**

Revision C and earlier revisions: Reserved. Revision D and later: Cold reset: 0000\_0000h. See  $F2x[1, 0]F0$  for register access information. This register is used by BIOS to specify the state of the ODT pins during DDR reads.  $F2x[1, 0]F4\_x180$  is used to control chip selects 0-3.  $F2x[1, 0]F4\_x181$  is used to control chip selects 4-7. See 2.8.9.5.7 [DRAM ODT Control] for more information.

Bits	Description
31:28	Reserved.
27:24	<b>RdOdtPatCs[7, 3]: read ODT pattern chip select [7, 3].</b> See $RdOdtPatCs[4, 0]$ .
23:20	Reserved.
19:16	<b>RdOdtPatCs[6, 2]: read ODT pattern chip select [6, 2].</b> See $RdOdtPatCs[4, 0]$ .
15:12	Reserved.
11:8	<b>RdOdtPatCs[5, 1]: read ODT pattern chip select [5, 1].</b> See $RdOdtPatCs[4, 0]$ .
7:4	Reserved.
3:0	<b>RdOdtPatCs[4, 0]: read ODT pattern chip select [4, 0].</b> Read-write. This field represents the state of ODT[3:0] pins when a read occurs to the specified chip select.

### **$F2x[1, 0]F4\_x[183:182]$ DCT ODT Control [High:Low] Registers**

Revision C and earlier revisions: Reserved. Revision D and later: Cold reset: 0000\_0000h. See  $F2x[1, 0]F0$  for

register access information. This register is used by BIOS to specify the state of the ODT pins during DDR writes. F2x[1, 0]F4\_x182 is used to control chip selects 0-3. F2x[1, 0]F4\_x183 is used to control chip selects 4-7. See 2.8.9.5.7 [DRAM ODT Control] for more information.

Bits	Description
31:28	Reserved.
27:24	<b>WrOdtPatCs[7, 3]: write ODT pattern chip select [7, 3].</b> See WrOdtPatCs[4, 0].
23:20	Reserved.
19:16	<b>WrOdtPatCs[6, 2]: write ODT pattern chip select [6, 2].</b> See WrOdtPatCs[4, 0].
15:12	Reserved.
11:8	<b>WrOdtPatCs[5, 1]: write ODT pattern chip select [5, 1].</b> See WrOdtPatCs[4, 0].
7:4	Reserved.
3:0	<b>WrOdtPatCs[4, 0]: write ODT pattern chip select [4, 0].</b> Read-write. This field represents the state of ODT[3:0] pins when a write occurs to the specified chip select.

### F2xF8 P-state Power Information Register

Revision C and earlier: Reserved.

Revision D and later: All fields are read-only.

Bits	Description
31:24	<b>PwrValue3.</b> This field applies to P-state 3. See PwrValue0.
23:16	<b>PwrValue2.</b> This field applies to P-state 2. See PwrValue0.
15:8	<b>PwrValue1.</b> This field applies to P-state 1. See PwrValue0.
7:0	<b>PwrValue0.</b> F2xF8[PwrValue0] and F2xFC[PwrDiv0] combine to specify the expected power draw of a single CPU core and 1/numCores of the Northbridge in P-state 0.

### F2xFC P-state Power Information Register 2

Revision C and earlier: Reserved.

Revision D and later: All fields are read-only.

Bits	Description
17:16	<b>PwrDiv4.</b> This field applies to P-state 4. See F2xF8[PwrValue0].
15:14	<b>PwrDiv3.</b> This field applies to P-state 3. See F2xF8[PwrValue0].
13:12	<b>PwrDiv2.</b> This field applies to P-state 2. See F2xF8[PwrValue0].
11:10	<b>PwrDiv1.</b> This field applies to P-state 1. See F2xF8[PwrValue0].
9:8	<b>PwrDiv0.</b> This field applies to P-state 0. See F2xF8[PwrValue0].
7:0	<b>PwrValue4.</b> This field applies to P-state 4. See F2xF8[PwrValue0].

### F2x10C Swap Interleaved Region Base/Limit Register

Reset: 0000 0000h. Revision C2 and earlier and revision D: Reserved. Revision C3 and E: Enables swapping a region below 16G with the same sized region located at the bottom of memory. This register is typically used to map addresses of a graphics frame buffer located below the sub-4GB IO hole to interleaved DRAM in low

memory, and is only necessary if the frame buffer normally maps to non-interleaved memory.

- The size of the swapped high region must be a integer multiple of 128M, defined to be {F2x10C[IntLvRegionBase],000b,000000h} to {F2x10C[IntLvRegionLimit],111b,FFFFFFh}.
- The size of the swapped region must be less than or equal to the alignment of F2x10C[IntLvRegionBase].
  - e.g. if F2x10C[IntLvRegionBase]=2h then size <=256MB and BIOS programs F2x10C[IntLvRegionLimit] <= F2x10C[IntLvRegionBase]+1h.
  - It is expected that BIOS may program F2x10C[IntLvRegionBase] to a value less than the base address of the graphics frame buffer if realignment is necessary to achieve a larger swap size.
- The location of the low region is defined to be 0000\_0000h to {F2x10C[IntLvRegionLimit]-F2x10C[IntLvRegionBase],111b,FFFFFFh}.
- The swapped region must be all DRAM. I.e. No IO hole.
- Channel interleaving must be enabled and the DCTs must be of unequal size.
- Swapping must not be enabled on more than one node, and F1x[1, 0][7C:40][DramBase] must be zero.
- See F2x110[DctSelIntLvEn]. See 2.8.11 [Memory Hoisting] for programming information. See 2.8.6 [Routing DRAM Requests].

Bits	Description
31:27	Reserved.
26:20	<b>IntLvRgnSize[33:27]: Interleave swap region size bits[33:27].</b> Read-write. Interleave swap region size [33:27].
19:18	Reserved.
17:11	<b>IntLvRgnLmtAddr[33:27]: Interleave swap region limit address bits[33:27].</b> Read-write. Interleave swap region limit address [33:27].
10	Reserved.
9:3	<b>IntLvRgnBaseAddr[33:27]: Interleave swap region base address bits[33:27].</b> Read-write. Interleave swap region base address [33:27].
2:1	Reserved.
0	<b>IntLvRgnSwapEn: Interleave region swap enable.</b> Read-write. 1=Enables swapping a region from the top of memory to the bottom of DRAM space.

### F2x110 DRAM Controller Select Low Register

Reset: 0000 0000h.

Bits	Description
31:11	<b>DctSelBaseAddr[47:27]: DRAM controller select base address bits[47:27].</b> Read-write. If the DCTs are unganged (based on DctGangEn), this delineates the address range of the two DCTs by specifying the base address of the upper address range.
10	<b>MemCleared: memory cleared.</b> Read-only. 1=Memory has been cleared since the last warm reset. This bit is set by MemClrInit. See MemClrInit below.
9	<b>MemClrBusy: memory clear busy.</b> Read-only. 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. See MemClrInit.
8	<b>DramEnable: DRAM enabled.</b> Read-only. 1=All of the used DCTs are initialized (see 2.8.9.6 [DRAM Device and Controller Initialization]) or have exited from self refresh (F2x[1, 0]90[ExitSel-fRef] transitions from 1 to 0).

7:6	<p><b>DctSelIntLvAddr: DRAM controller select channel interleave address bit.</b> Read-write. This specifies how interleaving is selected between the DCTs. In all cases, if the select function is low then DCT0 is selected; if the select function is high then DCT1 is selected. The select function is dependent on DctSelIntLvAddr and F1x124[DramIntlvEn] as follows:</p> <table border="1"> <thead> <tr> <th><u>DctSelIntLvAddr</u></th> <th><u>F1x124[DramIntlvEn]</u></th> <th><u>DCT Select Function</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>xxx</td> <td>Address bit 6.</td> </tr> <tr> <td>01b</td> <td>000b</td> <td>Address bit 12.</td> </tr> <tr> <td>01b</td> <td>001b</td> <td>Address bit 13.</td> </tr> <tr> <td>01b</td> <td>011b</td> <td>Address bit 14.</td> </tr> <tr> <td>01b</td> <td>111b</td> <td>Address bit 15.</td> </tr> <tr> <td>10b</td> <td>xxx</td> <td>Hash: exclusive OR of address bits[20:16, 6].</td> </tr> <tr> <td>11b</td> <td>xxx</td> <td>Hash<sup>1</sup>: exclusive OR of address bits[20:16, 9].</td> </tr> </tbody> </table> <p>1. This encoding is reserved for revision B and earlier. All other encodings are reserved.</p>	<u>DctSelIntLvAddr</u>	<u>F1x124[DramIntlvEn]</u>	<u>DCT Select Function</u>	00b	xxx	Address bit 6.	01b	000b	Address bit 12.	01b	001b	Address bit 13.	01b	011b	Address bit 14.	01b	111b	Address bit 15.	10b	xxx	Hash: exclusive OR of address bits[20:16, 6].	11b	xxx	Hash <sup>1</sup> : exclusive OR of address bits[20:16, 9].
<u>DctSelIntLvAddr</u>	<u>F1x124[DramIntlvEn]</u>	<u>DCT Select Function</u>																							
00b	xxx	Address bit 6.																							
01b	000b	Address bit 12.																							
01b	001b	Address bit 13.																							
01b	011b	Address bit 14.																							
01b	111b	Address bit 15.																							
10b	xxx	Hash: exclusive OR of address bits[20:16, 6].																							
11b	xxx	Hash <sup>1</sup> : exclusive OR of address bits[20:16, 9].																							
5	<p><b>DctDatIntLv: DRAM controller data interleave enable.</b> Read-write. 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. If ECC is enabled and the DCT is unganged (DctGangEn=0), DctDatIntLv should be enabled. See 2.12.2 [DRAM Considerations for ECC] for more information.</p>																								
4	<p><b>DctGangEn: DRAM controller ganging enable.</b> Read-write. 1=Both DCTs are ganged to form a single double-width DDR interface. 0=The DCTs operate independently. This also affects DCT configuration registers; see 2.8.1 [DCT Configuration Registers] and 2.8.5 [Ganged or Unganged Mode Considerations]. Note, if ganging is to be enabled, this bit must be set prior to programming any DCT registers.</p>																								
3	<p><b>MemClrInit: memory clear initialization.</b> Write only; reads as 0. 1=The node writes 0's to all locations of system memory attached to the node and sets the MemCleared bit. The memory clear operation is complete when MemCleared=1 and MemClrBusy=0. This command is ignored if MemClrBusy=1 when the command is received. Note: DramEnable must be set before setting MemClrInit. The memory prefetcher (see F2x11C) must be disabled before memory clear initialization and then can be re-enabled when MemCleared=1.</p>																								
2	<p><b>DctSelIntLvEn: DRAM controller interleave enable.</b> Read-write. 1=Channel interleave is enabled; DctSelIntLvAddr specifies which address bit is used to select between DCT0 and DCT1; this applies from the base system memory address of the node (specified by [The DRAM Base/Limit Registers] F1x[1, 0][7C:40]) to DctSelBaseAddr (if enabled). If the amount of memory connected to each of the DCTs is different, then channel interleaving may be supported across the address range that includes both DCTs, the top of which is specified by DctSelBaseAddr; the remainder of the address space, above DctSelBaseAddr, would then be allocated to only the DCT connected to the larger amount of memory, specified by DctSelHi.</p>																								
1	<p><b>DctSelHi: DRAM controller high select.</b> Read-write. If DctSelHiRngEn is set, this specifies which DCT receives accesses with addresses in the high range (greater than or equal to DctSelBaseAddr). 0=High addresses go to DCT0. 1=High addresses go to DCT1.</p>																								
0	<p><b>DctSelHiRngEn: DRAM controller select high range enable.</b> Read-write. 1=Enables addresses greater than or equal to DctSelBaseAddr[47:27] to be used to select between DCT0 and DCT1; DctSelHi specifies which DCT occupies the high range. Note: if DctGangEn=1, then this bit is not used.</p>																								

**F2x114 DRAM Controller Select High Register**

Reset: 0000 0000h.

Bits	Description
31:10	<b>DctSelBaseOffset[47:26]: DRAM controller select base offset address bits[47:26].</b> Read-write. When <b>F2x110[DctSelHiRngEn]=1</b> , this value is subtracted from the physical address of certain transactions before being passed to the DCT. See <a href="#">2.8.11.2 [DctSelBaseOffset Programming]</a> for programming information.
9:0	Reserved.

**F2x118 Memory Controller Configuration Low Register**

Fields in this register (bits[17:0]) indicate priority of request types. These are encoded as follows:

Low	01b
Medium	00b
High	10b
Variable	11b

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. <table> <tbody> <tr> <td>0000b = 80ns</td> <td>0100b = 400ns</td> <td>1000b = 720ns</td> <td>1100b = 1040ns</td> </tr> <tr> <td>0001b = 160ns</td> <td>0101b = 480ns</td> <td>1001b = 800ns</td> <td>1101b = 1120ns</td> </tr> <tr> <td>0010b = 240ns</td> <td>0110b = 560ns</td> <td>1010b = 880ns</td> <td>1110b = 1200ns</td> </tr> <tr> <td>0011b = 320ns</td> <td>0111b = 640ns</td> <td>1011b = 960ns</td> <td>1111b = 1280ns</td> </tr> </tbody> </table>	0000b = 80ns	0100b = 400ns	1000b = 720ns	1100b = 1040ns	0001b = 160ns	0101b = 480ns	1001b = 800ns	1101b = 1120ns	0010b = 240ns	0110b = 560ns	1010b = 880ns	1110b = 1200ns	0011b = 320ns	0111b = 640ns	1011b = 960ns	1111b = 1280ns
0000b = 80ns	0100b = 400ns	1000b = 720ns	1100b = 1040ns														
0001b = 160ns	0101b = 480ns	1001b = 800ns	1101b = 1120ns														
0010b = 240ns	0110b = 560ns	1010b = 880ns	1110b = 1200ns														
0011b = 320ns	0111b = 640ns	1011b = 960ns	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.																
19:18	Reserved.																
17:16	<b>MctPriScrub: scrubber priority.</b> Read-write. Reset: medium (00b).																
15:14	<b>MctPriTrace: trace-mode request priority.</b> Read-write. Reset: high (10b). This must be set to high.																
13:12	<b>MctPriIsoc: display refresh read priority.</b> Read-write. Reset: high (10b). See <a href="#">2.6.4.2.5 [Display Refresh And IFCM]</a> .																
11:10	<b>MctPriWr: default write priority.</b> Read-write. Reset: low (01b).																

9:8	<b>MctPriDefault: default non-write priority.</b> Read-write. Reset: medium (00b).
7:6	<b>MctPriIsocWr: IO write with the isoch bit set priority.</b> Read-write. Reset: medium (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> Read-write. Reset: high (10b). This does not apply to isochronous traffic that is classified as display refresh.
3:2	<b>MctPriCpuWr: CPU write priority.</b> Read-write. Reset: low (01b).
1:0	<b>MctPriCpuRd: CPU read priority.</b> Read-write. Reset: medium (00b).

### **F2x11C Memory Controller Configuration High Register**

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 or 3 cachelines ahead, on revision C and earlier revisions, or 1, 2, 3, 4, or 5 cachelines ahead, on revision D and later, 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.

For revision C and later revisions, 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  $F2x1B0[AdapPrefMissRatio]$  then the prefetch distance is reduced by  $F2x1B0[AdapPrefNegativeStep]$ .
- If the ratio of prefetch requests that cannot return data to total prefetch requests is less than  $F2x1B0[AdapPrefMissRatio]$  then the prefetch distance is increased by  $F2x1B0[AdapPrefPositiveStep]$ .
- 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.

- For revision B the following settings should be used:  

PrefThreeConf = 7	PrefTwoConf = 7	PrefOneConf = 2
PrefConfSat = 1	MctWrLimit = 10h	DctWrLimit = 0
- For revision C and later revisions the following settings should be used:  

PrefThreeConf = 6	PrefTwoConf = 3	PrefOneConf = 2
PrefConfSat = 0	DctWrLimit = 0	
- For revision C, the following setting should be used: MctWrLimit = 10h.
- For revision D and later, the following setting should be used:  
 If the DCTs are ungangned then MctWrLimit = 18h, else MctWrLimit = 10h.

Note: BIOS should enable prefetching by clearing  $F2x11C[PrefIoDis]$  and  $F2x11C[PrefCpuDis]$ .

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. Reset: 0. Setting this bit causes write bursting to be cancelled 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. 1=Causes write bursting to be cancelled and all outstanding writes to be flushed to DRAM when in the stop-grant state. This bit should be set to ensure writes are drained to DRAM before reset is asserted for the suspend-to-RAM state.
28	<b>PrefDramTrainMode: prefetch DRAM training mode.</b> Read-write; cleared-by-hardware. Reset: 0. 1=Enable DRAM training mode. Hardware clears this bit when the prefetch request limit is reached. Writing a zero to this bit clears the prefetch buffer and disables the DRAM training mode prefetcher. BIOS must write a zero to this bit after training is complete. This bit is valid only when $F2x[1, 0]94[BurstLength32]=0$ or when $F2x[1, 0]94[Ddr3Mode]=1$ . See 2.8.9.6 [Continuous Pattern Generation].
27:25	<b>PrefThreeConf: prefetch three-ahead confidence.</b> Read-write. Revision C and earlier revisions: Reset: 110b. Revision D and later: Reset: 100b. Confidence level required in order to prefetch three cachelines ahead (same encoding as PrefTwoConf below).



24:22	<b>PrefTwoConf: prefetch two-ahead confidence.</b> Read-write. Reset: 011b. Confidence level required in order to prefetch two cachelines ahead. 000b = 0 001b = 2 ... 111b = 14										
21:20	<b>PrefOneConf: prefetch one-ahead confidence.</b> Read-write. Reset: 10b. Confidence level required in order to prefetch one ahead (0 through 3).										
19:18	<b>PrefConfSat: prefetch confidence saturation.</b> Read-write. Reset: 00. Specifies the point at which prefetch confidence level saturates and stops incrementing. 00b = 15 01b = 7 10b = 3 11b = Reserved.										
17:16	<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.										
15	<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).										
14	<b>PrefIoFixStrideEn: Prefetch IO fixed stride enable.</b> Read-write. Reset: 0. 1=The prefetch stride for IO requests is fixed (non-alternating).										
13	<b>PrefIoDis: prefetch IO-access disable.</b> Read-write. Reset: 1. 1=Disables IO requests from triggering prefetch requests.										
12	<b>PrefCpuDis: prefetch CPU-access disable.</b> Read-write. Reset: 1. 1=Disables CPU requests from triggering prefetch requests.										
11:7	<b>MctPrefReqLimit: memory controller prefetch request limit.</b> Read-write. Reset: Revision C and earlier revisions: 1Fh (31). Revision D and later: 1Eh (30). Specifies the maximum number of outstanding prefetch requests allowed. See <a href="#">F3x78</a> for restrictions on this field.										
6:2	<b>MctWrLimit: memory controller write-burst limit.</b> Read-write. Reset: 1_1111b. Specifies the number of writes in the memory controller queue before they are burst into the DCTs. 00000b = 32. 00001b = 31. ... 11110b = 2. 11111b = Write bursting disabled.										
1:0	<b>DctWrLimit: DRAM controller write limit.</b> Read-write. Reset: 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. <table border="0"> <tr> <td><u>Revisions C and earlier revisions</u></td> <td><u>Revision D and later</u></td> </tr> <tr> <td>00b = 0</td> <td>00b = 0</td> </tr> <tr> <td>01b = 1</td> <td>01b = 2</td> </tr> <tr> <td>10b = 2</td> <td>10b = 4</td> </tr> <tr> <td>11b = no limit.</td> <td>11b = 8</td> </tr> </table>	<u>Revisions C and earlier revisions</u>	<u>Revision D and later</u>	00b = 0	00b = 0	01b = 1	01b = 2	10b = 2	10b = 4	11b = no limit.	11b = 8
<u>Revisions C and earlier revisions</u>	<u>Revision D and later</u>										
00b = 0	00b = 0										
01b = 1	01b = 2										
10b = 2	10b = 4										
11b = no limit.	11b = 8										

### F2x1B0 Extended Memory Controller Configuration Low Register

Revision B: Reserved. Revision C and later: The main function of this register is to control the memory prefetcher. See [F2x11C](#) for more details of the adaptive prefetch scheme.

Bits	Description										
31:28	<p><b>DcqBwThrotWm: dcq bandwidth throttle watermark.</b> Revision C and earlier revisions: Reserved. Revision D and later: Read-write. Reset: 3h. 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. BIOS should program this field based on the DRAM speed as follows:</p> <table> <thead> <tr> <th><u>Memory Speed</u></th> <th><u>Watermark</u></th> </tr> </thead> <tbody> <tr> <td>DDR667</td> <td>4h.</td> </tr> <tr> <td>DDR800</td> <td>5h.</td> </tr> <tr> <td>DDR1066</td> <td>6h.</td> </tr> <tr> <td>DDR1333h.</td> <td></td> </tr> </tbody> </table>	<u>Memory Speed</u>	<u>Watermark</u>	DDR667	4h.	DDR800	5h.	DDR1066	6h.	DDR1333h.	
<u>Memory Speed</u>	<u>Watermark</u>										
DDR667	4h.										
DDR800	5h.										
DDR1066	6h.										
DDR1333h.											
27:25	<p><b>PrefFiveConf: prefetch five-ahead confidence.</b> Revision C and earlier revisions: Reserved. Revision D and later: Read-write. Reset: 110b. Confidence level required in order to prefetch five cache-lines ahead. BIOS should program this to 111b.</p> <p>000b = 0 001b = 2 ... 111b = 14</p>										
24:22	<p><b>PrefFourConf: prefetch four-ahead confidence.</b> Revision C and earlier revisions: Reserved. Revision D and later: Read-write. Reset: 101b. Confidence level required in order to prefetch four cache-lines ahead. BIOS should program this to 111b.</p> <p>000b = 0 001b = 2 ... 111b = 14</p>										
21:20	Reserved.										
19:18	Reserved.										
17:15	<p><b>SpecPrefThreshold: speculative prefetch threshold.</b> Revision C and earlier revisions: Reserved. Revision D and later: Read-write. Reset: 111b. This field specifies the threshold for the per core hit counters. The counters increment on every hit and decrement on every miss. The counters saturate at 64. The L3 predictor predicts if the per core counter is greater than:</p> <table> <tbody> <tr> <td>000b = Always predict hit.</td> <td>100b = 32</td> </tr> <tr> <td>001b = 8</td> <td>101b = 40</td> </tr> <tr> <td>010b = 16</td> <td>110b = 48</td> </tr> <tr> <td>011b = 24</td> <td>111b = 56</td> </tr> </tbody> </table>	000b = Always predict hit.	100b = 32	001b = 8	101b = 40	010b = 16	110b = 48	011b = 24	111b = 56		
000b = Always predict hit.	100b = 32										
001b = 8	101b = 40										
010b = 16	110b = 48										
011b = 24	111b = 56										
14	<p><b>SpecPrefMis: speculative prefetch predict miss.</b> Revision C and earlier revisions: Reserved. Revision D and later: Read-write. Reset: 0. 1=The L3 predictor always predicts miss for L3 demand reads. This bit overrides the setting in SpecPrefThreshold.</p>										
13	<p><b>SpecPrefDis: speculative prefetch disable.</b> Revision C and earlier revisions: Reserved. Revision D and later: Read-write. Reset: 0b. 1=Disable the L3 speculative miss prefetcher.</p>										
12	<p><b>EnSplitDctLimits: split DCT write limits enable.</b> Revision C and earlier revisions: Reserved. Revision D and later: Read-write. Reset: 0. 1=The number of writes specified by F2x11C[DctWrLmt and MctWrLmt] is per DCT. 0=The number of writes specified by F2x11C[DctWrLmt and MctWrLmt] is total writes independent of DCT. Setting this bit also affects the encoding of F2x11C[DctWrLmt]. If the DCTs are unganged, BIOS should program this to 1; else leave this field at 0.</p>										
11	<p><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.</p>										

10:8	<b>CohPrefPrbLmt: coherent prefetch probe limit.</b> Read-write. Reset: 000b. Specifies the maximum number of probes that can be outstanding for memory prefetch requests. BIOS should program this to 001b unless the probe filter is enabled. If the probe filter is enabled, BIOS should leave this field at 000b. See 2.6.5.1 [Probe Filter]. 000b = Probing disabled for memory prefetch requests. 001b = 4 outstanding probes. 010b = 8 outstanding probes. 011b = 16 outstanding probes. 1xxb = reserved.
7:6	Reserved.
5:4	<b>AdapPrefNegativeStep: adaptive prefetch negative step.</b> Read-write. Reset: 00b. Specifies the step size that the adaptive prefetch scheme uses when decreasing the prefetch distance. BIOS should leave this field at default. 00b = 8/16 01b = 2/16 10b = 4/16 11b = 16/16
3:2	<b>AdapPrefPositiveStep: adaptive prefetch positive step.</b> Read-write. Reset: 00b. Specifies the step size that the adaptive prefetch scheme uses when increasing the prefetch distance. BIOS should leave this field at default. 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. 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. BIOS should program this to 01b. 00b = 1/16 01b = 2/16 10b = 4/16 11b = 8/16

### 3.6 Function 3 Miscellaneous Control Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention. See 2.11 [Configuration Space] for details about how to access this space.

#### F3x00 Device/Vendor ID Register

Reset: 1203 1022h.

Bits	Description
31:16	<b>DeviceID: device ID.</b> Read-only.
15:0	<b>VendorID: vendor ID.</b> Read-only.

#### F3x04 Status/Command Register

Reset: 0000 0000h, except bit[20]; see below.

Bits	Description
------	-------------

31:16	<b>Status.</b> Read-only. Bit[20] is set to indicate the existence of a PCI-defined capability block, if one exists.
15:0	<b>Command.</b> Read-only.

### F3x08 Class Code/Revision ID Register

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.

### F3x0C Header Type Register

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.

### F3x34 Capability Pointer Register

Reset: 0000 00??h.

Bits	Description
31:8	Reserved.
7:0	<b>CapPtr.</b> Read-only. Specifies the configuration-space offset of the capabilities pointer. If a capability block is enabled, this reads a F0h; otherwise it is 00h.

### F3x40 MCA NB Control Register

Reset: 0000 0000h. The machine check registers are used to configure the Machine Check Architecture (MCA) functions of the Northbridge (NB) hardware and to provide a method for the NB to report errors in a way compatible with MCA. All of the NB MCA registers, except [\[The MCA NB Configuration Register\] F3x44](#), are accessible through the MCA-defined MSR method, as well as through PCI configuration space. This register is also accessible through [MSR0000\\_0410](#). The NB control mask register is formatted identically to this register and is accessible through [MSRC001\\_0048](#) (see [MSRC001\\_00\[49:44\]](#)).

[F3x40](#) enables MCA reporting of each error checked by the NB. The global MCA error enables must also be set through [\[The Global Machine Check Exception Reporting Control Register \(MCG\\_CTL\)\] MSR0000\\_017B](#). The error enables in this register only affect error reporting through MCA. Actions which the NB may take in addition to MCA reporting are enabled through [\[The MCA NB Configuration Register\] F3x44](#).

Correctable and uncorrectable errors are logged in [\[The MCA NB Status Low Register\] F3x48](#), [\[The MCA NB Status High Register\] F3x4C](#), [\[The MCA NB Address Low Register\] F3x50](#), and [\[The DRAM Scrub Address High Register\] F3x60](#) as they occur, as specified by [F3x4C\[Over\]](#). Uncorrectable errors immediately result in a Machine Check exception. Correctable errors only increment a counter in [\[The NB Machine Check Misc \(Thresholding\) Registers\] F3x1\[78, 70, 68, 60\]](#), which may result in a Machine Check exception or a System Management Interrupt.

Bit	Description
31:28	Reserved.
27	<b>TblWlkDatErrEn: table walk data error enable.</b> Read-write. 1=Enables MCA reporting of uncorrectable errors in returned data from a DEV or GART table walk.
26	<b>NbArrayParEn: Northbridge array parity error reporting enable.</b> Read-write. 1=Enables MCA reporting of parity errors in the NB arrays.
25	<b>McaUsPwDatErrEn: MCA upstream data error enable.</b> Read-write. 1=Enables MCA reporting of upstream posted writes in which the link error bits indicate a data error.
24	<b>SyncPkt3En: link 3 sync packet error reporting enable.</b> Read-write. 1=Enables MCA reporting of link-defined sync error packets detected on link 3. The NB floods its outgoing links with sync packets after detecting a sync packet on an incoming link independent of the state of this bit.
23	<b>CrcErr3En: link 3 CRC error reporting enable.</b> Read-write. 1=Enables MCA reporting of CRC errors detected on link 3 (see the description of CRC Error in <a href="#">Table 92</a> ). The NB floods its outgoing links with sync packets after detecting a CRC error on an incoming link independent of the state of this bit.
22	<b>RtryHt3En: link 3 retry reporting enable.</b> Read-write. 1=Enables MCA reporting of retries on link 3.
21	<b>RtryHt2En: link 2 retry reporting enable.</b> Read-write. 1=Enables MCA reporting of retries on link 2.
20	<b>RtryHt1En: link 1 retry reporting enable.</b> Read-write. 1=Enables MCA reporting of retries on link 1.
19	<b>RtryHt0En: link 0 retry reporting enable.</b> Read-write. 1=Enables MCA reporting of retries on link 0.
18	<b>DramParEn: DRAM parity error reporting enable.</b> Read-write. 1=Enables MCA reporting of parity errors on the DRAM address or control signals.
17	<b>HtDataEn: link data error reporting enable.</b> Read-write. 1=Enables MCA reporting of packets with data errors detected on links.
16	<b>ProtEn: protocol error reporting enable.</b> Read-write. 1=Enables MCA reporting of protocol errors detected on links or in the L3 cache. When possible, this enable should be cleared before initiating a warm reset to avoid logging spurious errors due to RESET# signal skew.
15	<b>L3ArrayUCEn: L3 cache array uncorrectable error reporting enable.</b> Read-write. 1=Enables MCA reporting of uncorrectable errors in the L3 cache arrays.
14	<b>L3ArrayCorEn: L3 cache array correctable error reporting enable.</b> Read-write. 1=Enables MCA reporting of correctable errors in the L3 cache arrays.
13	<b>DevErrEn: DEV error reporting enable.</b> Read-write. 1=Enables MCA reporting of SVM DEV errors.
12	<b>WDTRptEn: watchdog timer error reporting enable.</b> Read-write. 1=Enables MCA reporting of watchdog timer errors. The watchdog timer checks for NB system accesses for which a response is expected but no response is received. See <a href="#">[The MCA NB Configuration Register] F3x44</a> for information regarding configuration of the watchdog timer duration. Note that 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> Read-write. 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	<b>GartTblWkEn: GART table walk error reporting enable.</b> Read-write. 1=Enables MCA reporting of GART cache table walks which encounter a GART PTE entry which is invalid.
9	<b>TgtAbortEn: target abort error reporting enable.</b> Read-write. 1=Enables MCA reporting of target aborts to a link. The NB returns an error response back to the requestor with any associated data all 1s independent of the state of this bit.
8	<b>MstrAbortEn: master abort error reporting enable.</b> Read-write. 1=Enables MCA reporting of master aborts to a link. The NB returns an error response back to the requestor with any associated data all 1s independent of the state of this bit.
7	<b>SyncPkt2En: link 2 sync packet error reporting enable.</b> Read-write. 1=Enables MCA reporting of link-defined sync error packets detected on link 2. The NB floods its outgoing links with sync packets after detecting a sync packet on an incoming link independent of the state of this bit.
6	<b>SyncPkt1En: link 1 sync packet error reporting enable.</b> Read-write. 1=Enables MCA reporting of link-defined sync error packets detected on link 1. The NB floods its outgoing links with sync packets after detecting a sync packet on an incoming link independent of the state of this bit.
5	<b>SyncPkt0En: link 0 sync packet error reporting enable.</b> Read-write. 1=Enables MCA reporting of link-defined sync error packets detected on link 0. The NB floods its outgoing links with sync packets after detecting a sync packet on an incoming link independent of the state of this bit.
4	<b>CrcErr2En: link 2 CRC error reporting enable.</b> Read-write. 1=Enables MCA reporting of CRC errors detected on link 2 (see the description of CRC Error in <a href="#">Table 92</a> ). The NB floods its outgoing links with sync packets after detecting a CRC error on an incoming link independent of the state of this bit.
3	<b>CrcErr1En: link 1 CRC error reporting enable.</b> Read-write. 1=Enables MCA reporting of CRC errors detected on link 1 (see the description of CRC Error in <a href="#">Table 92</a> ). The NB floods its outgoing links with sync packets after detecting a CRC error on an incoming link independent of the state of this bit.
2	<b>CrcErr0En: link 0 CRC error reporting enable.</b> Read-write. 1=Enables MCA reporting of CRC errors detected on link 0 (see the description of CRC Error in <a href="#">Table 92</a> ). The NB floods its outgoing links with sync packets after detecting a CRC error on an incoming link independent of the state of this bit.
1	<b>UECCEn: uncorrectable ECC error reporting enable.</b> Read-write. 1=Enables MCA reporting of DRAM uncorrectable ECC errors which are detected in the NB. If masked in MCI_CTL_MASK, the ECC error is not detected or logged.
0	<b>CECCEn: correctable ECC error reporting enable.</b> Read-write. 1=Enables MCA reporting of DRAM correctable ECC errors which are detected in the NB. If masked in MCI_CTL_MASK, the ECC error is detected and corrected, but not logged.



### F3x44 MCA NB Configuration Register

Reset: 0080 0000h. See [The Extended NB MCA Configuration Register] F3x180. Generally, it is expected that the fields of this register are programmed to the same value in all nodes (except for bit fields used for error injection, SubLinkSel, GenCrcErrByte1, GenCrcErrByte0, LdtLinkSel).

Bits	Description
31	<b>NbMcaLogEn: Northbridge MCA log enable.</b> Read-write. 1=Enables logging (but not reporting) of NB MCA errors even if MCA is not globally enabled.
30	<b>SyncOnDramAdrParErrEn: sync flood on DRAM address parity error enable.</b> Read-write. 1=Enables sync flood on detection of a DRAM address parity error. BIOS should set this bit to 1 for normal operation.
29	<b>DisMstAbtCpuErrRsp: master abort CPU error response disable.</b> Read-write. 1=Disables master abort reporting through the CPU MCA error-reporting banks.
28	<b>DisTgtAbtCpuErrRsp: target abort CPU error response disable.</b> Read-write. 1=Disables target abort reporting through the CPU MCA error-reporting banks.
27	<b>NbMcaToMstCpuEn: machine check errors to master CPU only.</b> Read-write. 1=NB MCA errors in a CMP device are reported only 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="#">MSRC000_04[0A:08]</a> , <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 field does not affect PCI-defined configuration space accesses to these registers, which are accessible from all cores. See 3.1 [Register Descriptions and Mnemonics] for a description of MSR space and 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. Notes: <ul style="list-style-type: none"> <li>• When the CPU which originated the request is known, it is stored in <a href="#">F3x4C[ErrCPU]</a>, regardless of the setting of NbMcaToMstCpuEn. See <a href="#">Table 94</a> for errors where ErrCPU is known.</li> <li>• If IO originated the request, then the error is reported to the NBC, regardless of the setting of NbMcaToMstCpuEn.</li> <li>• BIOS should set this bit to 1 in all processors.</li> </ul>
26	<b>CorrMcaExcEn: correctable error MCA exception enable.</b> Read-write. 1=A machine check exception is generated for correctable machine check errors that are enabled in <a href="#">F3x40</a> . This field is intended as a test and debug aid. It does not cause any status register changes, and the usual status overwrite and overflow rules remain in effect.
25	<b>DisPciCfgCpuErrRsp: PCI configuration CPU error response disable.</b> Read-write. 1=Disables generation of an error response to the core on detection of a master abort, target abort, or data error condition, and disables logging and reporting through the MCA error-reporting banks for PCI configuration accesses. Also, for NB WDT errors on PCI configuration accesses, this prevents sending an error response to the core, but does not affect logging and reporting of the NB WDT error. See <a href="#">F3x180[DisPciCfgCpuMstAbtRsp]</a> , which applies only to master aborts.
24	<b>IoRdDatErrEn: IO read data error log enable.</b> 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.





11:9	<p><b>WDTCntSel[2:0]: watchdog timer count select bits[2:0].</b> Read-write. Selects the count used by the watchdog timer. WDTCntSel is a 4-bit field composed of {F3x180[WDTCntSel[3]], F3x44[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> <tr> <td>0000b = 4095</td> <td>0100b = 255</td> <td>1000b = 8191</td> </tr> <tr> <td>0001b = 2047</td> <td>0101b = 127</td> <td>1001b = 16383</td> </tr> <tr> <td>0010b = 1023</td> <td>0110b = 63</td> <td>1010b - 1111b = reserved</td> </tr> <tr> <td>0011b = 511</td> <td>0111b = 31</td> <td></td> </tr> </table> <p>Note: 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>	0000b = 4095	0100b = 255	1000b = 8191	0001b = 2047	0101b = 127	1001b = 16383	0010b = 1023	0110b = 63	1010b - 1111b = reserved	0011b = 511	0111b = 31	
0000b = 4095	0100b = 255	1000b = 8191											
0001b = 2047	0101b = 127	1001b = 16383											
0010b = 1023	0110b = 63	1010b - 1111b = reserved											
0011b = 511	0111b = 31												
8	<p><b>WDTDis: watchdog timer disable.</b> Revisions C and earlier: Reset 0. Revision D and later: Cold reset 0. Read-write. 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 [The MCA NB Control Register] F3x40.</p>												
7	<p><b>IoErrDis: IO error response disable.</b> Read-write. 1=Disables setting either Error bit in link response packets to IO devices on detection of a target abort, master abort, or data error condition.</p>												
6	<p><b>CpuErrDis: CPU error response disable.</b> Read-write. 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. 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 F3x180[MstAbtChgToNoErrs] are both set, F3x180[MstAbtChgToNoErrs] takes precedence.</p>												
4	<p><b>SyncPktPropDis: sync packet propagation disable.</b> Read-write. 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. BIOS should set this bit to 1 for normal operation.</p>												
3	<p><b>SyncPktGenDis: sync packet generation disable.</b> Read-write. 1=Disables flooding of all outgoing links with sync packets when a CRC error is detected on an incoming link. BIOS should set this bit to 1 for normal operation. By default, sync packet generation for CRC errors is controlled through [The Link Control Registers] F0x[E4, C4, A4, 84].</p>												
2	<p><b>SyncOnUcEccEn: sync flood on uncorrectable ECC error enable.</b> Read-write. 1=Enables flooding of all links with sync packets on detection of an uncorrectable ECC error. BIOS should set this bit to 1 for normal operation.</p>												
1	<p><b>CpuRdDatErrEn: CPU read data error log enable.</b> Read-write. 1=Enables MCA logging and reporting of read responses and TgtDone responses which indicate master abort, target abort, or data error for CPU requests from this node. This bit should be clear if error response 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.</p>												
0	Reserved.												

**F3x48 MCA NB Status Low Register**

Cold reset: xxxx xxxhx.

Software is normally only allowed to write 0's to this register to clear the fields so subsequent errors may be logged. See [MSRC001\\_0015](#)[McStatusWrEn]. This register may be accessed through [\[The NB Machine Check Status Register \(MC4\\_STATUS\)\]](#) [MSR0000\\_0411](#) as well.

Bits	Description
31:24	<b>Syndrome[15:8]: syndrome bits 15:8 for ECC.</b> Read-write. Logs the upper eight syndrome bits when an ECC error is detected.
23:21	Reserved.
20:16	<b>ErrorCodeExt: extended error code.</b> Read-write. Logs the extended error code when an error is detected. See <a href="#">Table 93</a> for encoding.
15:0	<b>ErrorCode: error code.</b> Read-write. Logs an error code when an error is detected. See <a href="#">Table 93</a> for encoding.

Three types of errors are reported: TLB, memory, or bus errors.

Error Code	Error Code Type	Description
0000 0000 0001 TTLL	TLB	Errors in the GART TLB cache. 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

**Table 87: Error codes: transaction type**

TT	Transaction Type
00	Instruction
01	Data
10	Generic
11	Reserved

**Table 88: Error codes: cache level**

LL	Cache Level
00	Reserved
01	Level 1 (L1)
10	Level 2 (L2)

**Table 88: Error codes: cache level**

LL	Cache Level
11	Generic (LG; includes L3 cache)

**Table 89: Error codes: memory transaction type**

RRRR	Memory Transaction Type
0000	GEN: Generic. Includes scrub errors.
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 90: Error codes: participation processor**

PP	Participation Processor
00	Local node originated the request (SRC)
01	Local node responded to the request (RES)
10	Local node observed the error as a third party (OBS)
11	Generic

**Table 91: Error codes: memory or IO**

II	Memory or IO
00	Memory Access (MEM)
01	Reserved
10	IO Access (IO)
11	Generic (GEN)

**Table 92: NB error descriptions**

Error Type	Description	Control Bits (F3x40)
CRC Error	CRC error detected on link. If the link is in retry mode, this may indicate excessive link reconnect failures; see F0x[E4, C4, A4, 84][CrcErr, LinkFail, CrcFloodEn].  The NB floods its outgoing links with sync packets after detecting a CRC error on an incoming link independent of the state of the control bits.	CrcErr0En, CrcErr1En, CrcErr2En, CrcErr3En
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.	SyncPkt0En, SyncPkt1En, SyncPkt2En, SyncPkt3En
Master Abort	Master abort seen as result of link operation. Reasons for this error include requests to non-existent addresses, and requesting extended addresses while extended mode disabled (see F0x[E4, C4, A4, 84][Addr64BitEn]). The NB returns an error response back to the requestor with any associated data all 1s independent of the state of the control bit.	MstrAbortEn
Target Abort	Target abort seen as result of link operation. The NB returns an error response back to the requestor with any associated data all 1s independent of the state of the control bit.	TgtAbortEn
GART Error	GART cache table walk encountered a GART PTE entry which was invalid.	GartTblWkEn
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
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
ECC Error	DRAM ECC error detected in the NB.	CECCEn, UECCEn
DEV Error	SVM DEV error detected.	DevErrEn

**Table 92: NB error descriptions**

Error Type	Description	Control Bits (F3x40)
Link Data Error	Data error detected on link.  If enabled for reporting and the request is sourced from a CPU, 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 CPU, this could allow error containment to a scope smaller than the entire system.)	HtDataEn, McaUsPwDatErrEn,
Protocol Error	Protocol error detected by link, L3, or probe filter. These errors are distinguished from each other by the value in <a href="#">MSR0000_0412[ErrAddr]</a> . See <a href="#">Table 96</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.  For link protocol errors, ensure that the error is not due to failure or reset at the far end of link or from transmission corruption, indicated by CRC error. The enable for this error should be cleared before initiating a warm reset to avoid logging spurious errors due to RESET# signal skew.	ProtEn
NB Array Error	A parity error was detected in the NB internal arrays.	NbArrayParEn
DRAM Parity Error	A parity error was detected on the DRAM address or control signals.	DramParEn
Link Retry	A transmission error occurred on the link; the IO link Error Retry Protocol is executed. Retry may have been initiated by either end of the link.	RtryHt0En, RtryHt1En, RtryHt2En, RtryHt3En
GART Table Walk Data Error	An uncorrectable error was found in data returned from a GART table walk.	TblWlkDatErrEn
DEV Table Walk Data Error	An uncorrectable error was found in data returned from a DEV table walk.	TblWlkDatErrEn
Probe Filter Error	An ECC error was detected in the probe filter directory. See <a href="#">2.6.5.1.1 [Probe Filter Errors]</a> for more details.	
L3 Cache Data Error	ECC error detected in L3 cache data. A sync flood occurs.	L3ArrayCorEn, L3ArrayUCEn
L3 Cache Tag Error	Error detected in L3 cache tag. A sync flood occurs.  The subcache, index, and way are logged. See <a href="#">Table 94</a> footnotes for details.	L3ArrayCorEn, L3ArrayUCEn
L3 Cache LRU Error	Error detected in LRU parity bits. This is a non-fatal error which has no impact on any program execution; LRU state is reset. The cache index is captured for thresholding purposes.	

The NB is capable of reporting the following errors

**Table 93: NB error signatures, part 1**

Error Type	Error Threshold Group	20:16 Ext. Error	Error Code (see F3x48 for encoding)					
			Type	10:9 PP	8 T	7:4 RRRR	3:2 II/TT	1:0 LL
Reserved.	-	0_0000	-	-	-	-	-	-
CRC Error	Link	0_0001	BUS	OBS	0	GEN	GEN	LG
Sync Error		0_0010	BUS	OBS	0	GEN	GEN	LG
Mst Abort		0_0011	BUS	SRC/OBS	0	RD/WR	MEM/IO <sup>1</sup>	LG
Tgt Abort		0_0100	BUS	SRC/OBS	0	RD/WR	MEM/IO <sup>1</sup>	LG
GART Error		0_0101	TLB	-	-	-	GEN	LG
RMW Error		0_0110	BUS	OBS	0	GEN	IO	LG
WDT Error		0_0111	BUS	GEN	1	GEN	GEN	LG
ECC Error		DRAM	0_1000	BUS	SRC/RES	0	RD/WR	MEM
DEV Error	Link	0_1001	BUS	SRC/OBS	0	RD/WR	MEM/IO	LG
Link Data Error		0_1010	BUS	SRC/OBS	0	RD/WR/ DWR	MEM/IO	LG
Protocol Error	Link/ <sup>2</sup>	0_1011	BUS	OBS	0	GEN	GEN	LG
NB Array Error	-	0_1100	BUS	OBS	0	GEN	GEN	LG
DRAM Parity Error	DRAM	0_1101	BUS	OBS	0	GEN	MEM	LG
Link Retry	Link	0_1110	BUS	OBS	0	GEN	GEN	LG
GART Table Walk Data Error		0_1111	TLB	-	-	-	GEN	LG
DEV Table Walk Data Error		0_1111	BUS	OBS	0	GEN	MEM	LG
Probe Filter Error	L3 Cache	1_1111	MEM	-	-	RD/Evict /Snoop /GEN	GEN	LG
L3 Cache Data Error	L3 Cache	1_1100	MEM	-	-	RD/Evict /Snoop /GEN	GEN	LG
L3 Cache Tag Error		1_1101	MEM	-	-	RD/Evict /Snoop /GEN	GEN	LG
L3 Cache LRU Error		1_1110	MEM	-	-	RD/Evict /Snoop /GEN	GEN	LG

- Indicates the type of link attached to the reporting NB, not the instruction type. MEM indicates coherent link, IO indicates IO link.
- Error thresholding group is Link if link protocol error, none if L3 or probe filter protocol error.



**Table 94: NB error signatures, part 2**

Error Type	F3x4C settings								
	29 UC	26 AddrV	25 PCC	Syndrome Valid	14 CECC	13 UECC	8 Scrub	7:4 LDT Link	3:0 Err CPU
CRC Error	1	0	1	-	0	0	0	Y	-
Sync Error	1	0	1	-	0	0	0	Y	-
Mst Abort	1	1	If CPU source	-	0	0	0	Y	Y
Tgt Abort	1	1	If CPU source	-	0	0	0	Y	Y
GART Error	1	1	If CPU source	-	0	0	0	-	Y
RMW Error	1	1	0	-	0	0	0	Y	-
WDT Error	1	0 <sup>1</sup>	1	-	0	0	0	-	-
ECC Error	If multi-symbol <sup>8</sup>	1	If multi-symbol <sup>8</sup> and CPU source	15:0	If not multi-symbol <sup>8</sup>	If multi-symbol <sup>8</sup>	1/0	-	-
DEV Error	1	1	0	-	0	0	0	Y	-
Link Data Error	1	1	If En and CPU source	-	0	0	0	Y	-
Protocol Error	1	1/0 <sup>2</sup>	1	-	0	0	0	Y <sup>3</sup>	-
NB Array Error	1	1 <sup>4</sup>	1	-	0	0	0	-	-
DRAM Parity Error	1	0	1	-	0	0	0	-	-
Link Retry <sup>9</sup>	0	0	0	-	0	0	0	Y	-
GART Table Walk Data Error	1	1	If CPU source	-	0	0	0	-	If CPU source

1. See Table 99, “MCA NB Address Low Register encoding for Watchdog Timer Errors,”
2. See Table 96, “MCA NB Address Low Register encoding Protocol Errors,”
3. Link identified only if link protocol error. See entry in Table 92 for details.
4. See Table 97, “MCA NB Address Low Register encoding for NB Array Errors,”
5. See Table 98, “MCA NB Address Register encoding for L3 Array Errors,”
6. This field contains the L3 way in error. F3x4C[McaStatSubCache] contains the subcache number.
7. Depends on Memory Transaction Type (Table 89); valid if non-zero.
8. x4 Chipkill ECC is always enabled in ganged mode. See F3x180[EccSymbolSize].
9. Retries initiated by either side of the link are logged.

Table 94: NB error signatures, part 2

Error Type	F3x4C settings								
	29 UC	26 AddrV	25 PCC	Syndrome Valid	14 CECC	13 UECC	8 Scrub	7:4 LDT Link	3:0 Err CPU
DEV Table Walk Data Error	1	1	0	-	0	0	0	-	-
Probe Filter Error	0	1 <sup>5</sup>	0	15:0	If single-bit	If multi-bit	1/0	Y <sup>6</sup>	Y <sup>7</sup>
L3 Cache Data Error	If UECC	1 <sup>5</sup>	If UC and not Scrub	15:0	If single-bit	If multi-bit	1/0	Y <sup>6</sup>	Y <sup>7</sup>
L3 Cache Tag Error									
L3 Cache LRU Error	0	1 <sup>5</sup>	0	-	0	0	0	-	-
<ol style="list-style-type: none"> <li>See Table 99, “MCA NB Address Low Register encoding for Watchdog Timer Errors,”</li> <li>See Table 96, “MCA NB Address Low Register encoding Protocol Errors,”</li> <li>Link identified only if link protocol error. See entry in Table 92 for details.</li> <li>See Table 97, “MCA NB Address Low Register encoding for NB Array Errors,”</li> <li>See Table 98, “MCA NB Address Register encoding for L3 Array Errors,”</li> <li>This field contains the L3 way in error. F3x4C[McaStatSubCache] contains the subcache number.</li> <li>Depends on Memory Transaction Type (Table 89); valid if non-zero.</li> <li>x4 Chipkill ECC is always enabled in ganged mode. See F3x180[EccSymbolSize].</li> <li>Retries initiated by either side of the link are logged.</li> </ol>									

### F3x4C MCA NB Status High Register

Cold reset: xxxx xxxh.

This register may be accessed through [The NB Machine Check Status Register (MC4\_STATUS)] MSR0000\_0411 as well.

Software is normally only allowed to write 0's to this register to clear the fields so subsequent errors may be logged. See MSRC001\_0015[McStatusWrEn].

Bits	Description
31	<b>Val: error valid.</b> Read-write; set-by-hardware. 1=This bit indicates that a valid error has been detected. This bit should be cleared to 0 by software after the register has been read.
30	<b>Over: error overflow.</b> Read-write; set-by-hardware. 1=An error was detected while the valid bit (Val) of this register was set; at least one error was not logged. The machine check mechanism handles the contents of MCI_STATUS during overflow as outlined in section 2.12.1.2.2 [Error Overflow].
29	<b>UC: error uncorrected.</b> Read-write; set-by-hardware. 1=The error was not corrected by hardware.
28	<b>En: error enable.</b> Read-write; set-by-hardware. 1=The MCA error reporting is enabled for this error in the MCA Control Register.

27	<b>MiscV: miscellaneous error register valid.</b> Read-only. 1=The error currently logged in the NB MCA registers includes valid information in [The NB Machine Check Misc (Thresholding) Registers] F3x1[78, 70, 68, 60].						
26	<b>AddrV: error address valid.</b> Read-write; set-by-hardware. 1=The address saved in the address register is the address where the error occurred.						
25	<b>PCC: processor context corrupt.</b> Read-write; set-by-hardware. 1=The state of the processor may be corrupted by the error condition. Reliable restarting might not be possible.						
24	<b>ErrCpuVal: error cpu valid.</b> Revision C and earlier: Reserved. Revision D and later: Read-write; set-by-hardware. 1=The ErrCpu[3:0] field is valid.						
23	Reserved.						
22:15	<b>Syndromel[7:0]: syndrome bits [7:0] for ECC.</b> Read-write. Logs the lower eight syndrome bits when an ECC error is detected.						
14	<b>CECC: correctable ECC error.</b> Read-write; set-by-hardware. 1=The error was a correctable ECC error.						
13	<b>UECC: uncorrectable ECC error.</b> Read-write; set-by-hardware. 1=The error was an uncorrectable ECC error.						
12	Reserved.						
11:10	<b>McaStatSubCache: L3 subcache in error.</b> Indicates the number of the L3 subcache associated with the error. This field is only valid when an L3 error is recorded.						
9	<b>SubLink: sublink or DRAM channel.</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. For DRAM parity errors, this bit indicates which channel the error was associated with. <table border="0" style="margin-left: 40px;"> <tr> <td style="padding-right: 40px;"><u>DRAM Channel</u></td> <td><u>Sublink</u></td> </tr> <tr> <td>0 = Channel A</td> <td>0 = Bits [7:0]</td> </tr> <tr> <td>1 = Channel B</td> <td>1 = Bits [15:8]</td> </tr> </table>	<u>DRAM Channel</u>	<u>Sublink</u>	0 = Channel A	0 = Bits [7:0]	1 = Channel B	1 = Bits [15:8]
<u>DRAM Channel</u>	<u>Sublink</u>						
0 = Channel A	0 = Bits [7:0]						
1 = Channel B	1 = Bits [15:8]						
8	<b>Scrub: error found by DRAM scrubber.</b> Read-write; set-by-hardware. 1=The error was found by the DRAM scrubber.						
7:4	<b>LDTLink[3:0].</b> Read-write; set-by-hardware. For errors associated with a link, this field indicates which link was associated with the error. LDTLink[3] = Error associated with link 3. LDTLink[2] = Error associated with link 2. LDTLink[1] = Error associated with link 1. LDTLink[0] = Error associated with link 0.  For L3 cache errors, this field indicates the L3 way in error, and McaStatSubCache contains subcache number.						
3:0	<b>ErrCpu[3:0]: error associated with core.</b> Read-write; set-by-hardware. This field indicates which core within the node is associated with the error. Revision C and earlier: ErrCpu[3] = Error associated with core 3. ErrCpu[2] = Error associated with core 2. ErrCpu[1] = Error associated with core 1. ErrCpu[0] = Error associated with core 0. Revision D and later: ErrCpu[3:0] is an encoded value where 0h=core 0, 1h=core 1...5h=core 5. All other values are reserved. This field is valid only when ErrCpuVal=1.						

**F3x50 MCA NB Address Low Register**

Cold reset: xxxx xxxhx. F3x50 maps the lower half of [The NB Machine Check Address Register (MC4\_ADDR)] MSR0000\_0412, and F3x54 maps the upper half. MC4\_ADDR carries supplemental information associated with a machine check error, generally the address being accessed.

Bits	Description
31:1	<b>ErrAddr[31:1]: Error Address Bits[31:1].</b> Read-write. See the tables below for the encoding. ErrAddr[47:1] = {F3x54[ErrAddr[47:32]], F3x50[ErrAddr[31:1]]}.
0	Reserved.

The register format depends on the type of error being logged:

- Protocol errors contain the error reason code and, for revision D and later, may contain the physical address. They are formatted according to [Table 96](#).
- NB array errors indicate the array in error, and are formatted according to [Table 97](#).
- L3 array errors store the physical address which caused the error, and are formatted according to [Table 98](#).
- NB watchdog timer errors depend on the mode selected by F3x180[McaLogErrAddrWdtErr], and the format is indicated by F3x4C[AddrV]. If F3x4C[AddrV] is indicated, errors are formatted according to [Table 95](#). If F3x4C[AddrV] is not indicated, errors are formatted according to [Table 99](#).
- All other NB errors which indicate F3x4C[AddrV] are formatted according to [Table 95](#).

**Table 95: Default MCA NB Address Register default encoding**

ErrAddr bits	Description
47:1	Physical address bits 47:1
0	Reserved

**Table 96: MCA NB Address Low Register encoding Protocol Errors**

ErrAddr bits	Protocol Error Type	Description
47:6	-	If F3x4C[AddrV] set, contains physical address bits 47:6, else reserved.

**Table 96: MCA NB Address Low Register encoding Protocol Errors**

ErrAddr bits	Protocol Error Type	Description
5:1	0_0000b	Link: SRQ Read Response without matching request
	0_0001b	Link: Probe Response without matching request
	0_0010b	Link: TgtDone without matching request
	0_0011b	Link: TgtStart without matching request
	0_0100b	Link: Command buffer overflow
	0_0101b	Link: Data buffer overflow
	0_0110b	Link: Link retry packet count acknowledge overflow
	0_0111b	Data command in the middle of a data transfer
	0_1000b	Link: Link address extension command followed by a packet other than a command with address.
	0_1001b	Link: A specific coherent-only packet from a CPU was issued to an IO link. This may be caused by software which addresses page table structures in a memory type other than cacheable WB-DRAM without properly configuring <a href="#">MSRC001_0015</a> [Tlb-CacheDis]. This may occur, for example, when page table structure addresses are above top of memory. In such cases, the NB will generate an MCE if it sees a mismatch between the memory operation generated by the core and the link type. See <a href="#">2.9.3.1.2 [Determining The Access Destination for CPU Accesses]</a> .
	0_1010b	Link: A command with invalid encoding was received. This error occurs when: (1) 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 or (2) any illegal command is received in which the CRC is correct while in retry mode.
	0_1011b	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).
	1_0000b	L3: Request gets multiple hits in L3
	1_0001b	L3: Probe access gets multiple hits in L3
	1_0010b	L3: Request queue overflow
	1_0011b	L3: WrVicBlk hit incompatible L3 state
	1_0100b	L3: CIVicBlk hit incompatible L3 state
	1_1000b	PF: Directed probe miss
	1_1001b	PF: Directed probe clean hit
	1_1010b	PF: VicBlkM hit inconsistent directory state 'O S'.
1_1011b	PF: VicBlkE hit inconsistent directory state 'O S'.	
1_1100b	PF: Reserved.	
1_1101b	PF: L3 lookup response without a matching PFQ entry.	
1_1110b	PF: L3 update data read request without a matching PFQ entry.	
5:1	1_1111b	PF: Reserved
0	-	Reserved

**Table 97: MCA NB Address Low Register encoding for NB Array Errors**

<b>ErrAddr bits</b>	<b>Array Code</b>	<b>Description</b>
39:6	-	Reserved
5:1	0_0000b	SRA: System request address
	0_0001b	SRD: System request data
	0_0010b	SPB: System packet buffer
	0_0011b	MCD: Memory controller data
	0_0100b	MPB: Memory packet buffer
	0_0101b	LPB0: Link 0 packet buffer
	0_0110b	LPB1: Link 1 packet buffer
	0_0111b	LPB2: Link 2 packet buffer
	0_1000b	LPB3: Link 3 packet buffer
	0_1001b	MPBC: Memory controller command packet buffer
	0_1010b	MCDBM: Memory controller byte mask
	0_1011b	MCACAM: Memory controller address array
	0_1100b	DMAP: Extended DRAM address map
	0_1101b	MMAP: Extended MMIO address map
	0_1110b	X86MAP: Extended PCI/IO address map
	0_1111b	CFGMAP: Extended config address map
	1_0000b	LPS0: Link 0 packet state buffer
	1_0001b	LPS1: Link 1 packet state buffer
	1_0010b	LPS2: Link 2 packet state buffer
	1_0011b	LPS3: Link 3 packet state buffer
	1_0100b	RHB0: Link 0 retry history buffer
	1_0101b	RHB1: Link 1 retry history buffer
	1_0110b	RHB2: Link 2 retry history buffer
	1_0111b	RHB3: Link 3 retry history buffer
	1_1000b	SRIMCTRTE: SRI/MCT extended routing table
	1_1001b	LN0LN1RTE: Link 0/1 extended routing table
	1_1010b	LN2LN3RTE: Link 2/3 extended routing table
	1_1011b	GART: GART array
1_1100b	DEV: DEV array	
0	-	Reserved

**Table 98: MCA NB Address Register encoding for L3 Array Errors**

Error Type	Memory Transaction Type (RRRR; Table 89)	ErrAddr bits	Description
Data Error	RD, Evict	47:2	Physical address
		16:6 <sup>1</sup>	Cache index
	Snoop	47:6	Cache line address requested
		16:6 <sup>1</sup>	Cache index
		5:4	Index of the critical octword within the cache line
		3:2	Index of the octword in error within the cache line (normally occupies bits 5:4 in physical address)
GEN	16:6 <sup>1</sup>	Cache index	
Tag or LRU error	RD, Evict, Snoop	47:6	Physical address contained in the tag (may not match the address requested)
	GEN	16:6 <sup>1</sup>	Cache index

1. For 2MB subcaches, ErrAddr is bits 16:6. For 1MB subcaches, ErrAddr is bits 15:6. See [F3x1C4\[L3SubCacheSize\[3:0\]\]](#).

**Table 99: MCA NB Address Low Register encoding for Watchdog Timer Errors**

ErrAddr bits	Description
39:36	<b>System Response Count.</b> This field records unspecified, implementation-specific information.
35:31	<b>Wait Code</b> records unspecified, implementation-specific information (all zeroes means no waiting condition).
30	<b>Wait for Posted Write.</b>
29:27	<b>Destination Node</b> records the Node ID of the node addressed by the transaction.
26:25	<b>Destination Unit</b> is one of: 00 = Core 01 = GART or DEV Table Walker 10 = Memory Controller 11 = Host
24:22	<b>Source Node</b> records the Node ID of the node originating the transaction.
21:20	<b>Source Unit</b> (same encoding as Destination Unit)
19:15	<b>Source Pointer.</b> Identifies link as a crossbar source: 000xxb = GART or DEV Table Walker 001NNb = CPU number NN 010xxb = Memory controller 011xxb = Reserved 1HHxNb = Link HH, sublink N (where N=0b for ganged links)



**Table 99: MCA NB Address Low Register encoding for Watchdog Timer Errors**

ErrAddr bits	Description
14:11	<b>SRQ Entry State</b> records unspecified, implementation-specific information (all zeroes means idle).
10:7	<b>Op Type</b> records unspecified, implementation-specific information (all zeroes means normal).
6:1	<b>Link Command.</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 <i>HyperTransport™ I/O Link Specification</i> .

**F3x54 MCA NB Address High Register**

Cold reset: 0000 xxxxh. F3x54 maps the upper half of [The NB Machine Check Address Register (MC4\_ADDR)] MSR0000\_0412, and F3x50 maps the lower half.

Bits	Description
31:16	Reserved.
15:0	<b>ErrAddr[47:32]: Error Address Bits[47:32].</b> Read-write. See F3x50[ErrAddr] for details.

**F3x58 Scrub Rate Control Register**

Reset: 0000 0000h. This register specifies the ECC sequential scrubbing rate for lines of memory and cache. See 2.6.6 [Memory Scrubbers]. The scrub rate is specified as the time between successive scrub events. A scrub event occurs when a line is checked for errors; the size of the line that is checked varies based on the block (see field descriptions). Each of these fields is defined as follows:

Bits	Scrub Rate	Bits	Scrub Rate	Bits	Scrub Rate
00h	Disable sequential scrubbing	08h	5.12 us	10h	1.31 ms
01h	40 ns <sup>1</sup>	09h	10.2 us	11h	2.62 ms
02h	80 ns <sup>1</sup>	0Ah	20.5 us	12h	5.24 ms
03h	160 ns <sup>1</sup>	0Bh	41.0 us	13h	10.49 ms
04h	320 ns <sup>1</sup>	0Ch	81.9 us	14h	20.97 ms
05h	640 ns	0Dh	163.8 us	15h	42 ms
06h	1.28 us	0Eh	327.7 us	16h	84 ms
07h	2.56 us	0Fh	655.4 us		All others - reserved.

1. This setting is not supported except as a DRAM scrub rate when no other memory accesses are being performed.

Scrub rates are a platform consideration. See 2.12.1.7 [Scrub Rate Considerations].

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

$$\text{Time} = (\text{memory size in bytes}/64) * (\text{Scrub Rate}).$$

For example, if a node contains 1GB of system memory and DramScrub=5.24ms, then all of the system memory of the node is scrubbed about once every 24 hours.

Bits	Description
31:29	Reserved.
28:24	<b>L3Scrub: L3 cache scrub rate.</b> Read-write. Read-write. Specifies time between 64-byte scrub events. Note: the L3 scrubber should be disabled (L3Scrub=00h) if F3xE8[L3Capable]=0. See 2.6.5.1 [Probe Filter] for additional requirements.
23:21	Reserved.
20:16	<b>DcacheScrub: data cache scrub rate.</b> Read-write. Specifies time between 64-bit scrub events. BIOS must not set this scrub rate to less than 1.28 us.
15:13	Reserved.
12:8	<b>L2Scrub: L2 cache scrub rate.</b> Read-write. Read-write. Specifies time between 64-byte scrub events. BIOS must not set this scrub rate to less than 1.28 us.
7:5	Reserved.
4:0	<b>DramScrub: DRAM scrub rate.</b> Read-write. Specifies time between 64-byte scrub events. See F3x5C and F3x60. See 2.6.5.1 [Probe Filter] for additional requirements.

### F3x5C DRAM Scrub Address Low Register

Reset: see field definitions. 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 F3x5C[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 [The MCA NB Control Register] F3x40).

Bits	Description
31:6	<b>ScrubAddrLo: DRAM scrubber address bits[31:6].</b> Read; write to initialize; updated by hardware. Reset: 0. The DRAM scrubber address consists of {F3x60[ScrubAddrHi], F3x5C[ScrubAddrLo]}; it 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 [The DRAM Base/Limit Registers] F1x[1, 0][7C:40] prior to enabling sequential scrubbing through F3x58[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 F1x[1, 0][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. 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 <a href="#">F3x58[DramScrub]</a> ). 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. See <a href="#">2.6.5.1 [Probe Filter]</a> for additional requirements.

### F3x60 DRAM Scrub Address High Register

Reset: see field definitions.

Bits	Description
31:16	Reserved.
15:0	<b>ScrubAddrHi: DRAM scrubber address bits[47:32].</b> Read; write to initialize; updated by hardware. Reset: 0. See <a href="#">F3x5C[ScrubAddrLo]</a> .

### F3x64 Hardware Thermal Control (HTC) Register

See [2.10.3.1 \[PROCHOT\\_L and Hardware Thermal Control \(HTC\)\]](#) for information on HTC. F3x64 is not accessible if [\[The Northbridge Capabilities Register\] F3xE8\[HTC capable\]=0](#). F3x64 should only be programmed on internal node 0 ([F3xE8\[IntNodeNum\]=00b](#)) if [F3xE8\[MultiNodeCpu\]=1](#).

Bits	Description
31	Reserved.
30:28	<b>HtcPstateLimit: HTC P-state limit select.</b> Read-write. Reset state varies by product. 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 <a href="#">MSRC001_0061[PstateMaxVal]</a> + <a href="#">F4x15C[NumBoostStates]</a> . See <a href="#">2.10.3.1 [PROCHOT_L and Hardware Thermal Control (HTC)]</a> and <a href="#">2.4.2.1.2.2 [Hardware P-state Numbering]</a> .
27:24	<b>HtcHystLmt: HTC hysteresis.</b> Read-write. Reset state varies by product. The processor exits the HTC-active state when Tctl is less than HtcTmpLmt minus HtcHystLmt. The encoding is $0.5 * \text{HtcHystLmt}$ , ranging from 0.0 Tctl to 7.5 Tctl.
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="#">[The Reported Temperature Control Register] F3xA4</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 state varies by product. The processor enters the HTC-active state when Tctl reaches or exceeds the value of this register. The encoding is $52.0 + (0.5 * \text{HtcTmpLmt})$ , ranging from 52.0 Tctl to 115.5 Tctl.
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="#">[The Thermal Local Vector Table Entry] APIC330</a> of each core when the active P-state limit in <a href="#">[The P-State Current Limit Register] 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 above.
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. 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. 1=HTC is enabled; the processor is capable of entering the HTC-active state. For products that support NB P-states, BIOS must program <a href="#">MSRC001_001F[GfxNbPstateDis]=1</a> prior to setting HtcEn. BIOS should set this bit.

### F3x68 Software Thermal Control (STC) Register

Reset: ?000 0000h. See [2.10.3.2 \[Software Thermal Control \(STC\)\]](#) for information on STC. F3x68 is not accessible if [\[The Northbridge Capabilities Register\] F3xE8\[HTC capable\]=0](#). F3x68 should only be programmed on internal node 0 (F3xE8[IntNodeNum]=00b) if [F3xE8\[MultiNodeCpu\]=1](#).

Bits	Description
31	Reserved.
30:28	<b>StcPstateLimit: STC P-state limit select.</b> Read-write. Reset state varies by product. Specifies the P-state limit of all cores when in the STC-active state. This field uses hardware P-state numbering and is not changed on a write if the value written is greater than <a href="#">MSRC001_0061[PstateMaxVal]</a> + <a href="#">F4x15C[NumBoostStates]</a> . See <a href="#">2.10.3.2 [Software Thermal Control (STC)]</a> and <a href="#">2.4.2.1.2.2 [Hardware P-state Numbering]</a> .
27:24	<b>StcHystLmt: STC hysteresis.</b> Read-write. The processor exits the STC thermal zone when Tctl drops to StcTmpLmt minus StcHystLmt. Revision B: The encoding is $0.5 * \text{StcHystLmt}$ , ranging from 0.0 Tctl to 7.5 Tctl. Revision C and later revisions: The encoding is $1.0 * \text{StcHystLmt}$ , ranging from 0.0 Tctl to 15.0 Tctl.
23	<b>StcSlewSel: STC slew-controlled temperature select.</b> Read-write. Reset: 0. 1=STC logic is driven by the slew-controlled temperature, Tctl, specified in <a href="#">[The Reported Temperature Control Register] F3xA4</a> . 0=STC logic is driven by the measured control temperature with no slew controls.
22:16	<b>StcTmpLmt: STC temperature limit.</b> Read-write. The processor enters the STC thermal zone when Tctl exceeds the value specified by this register. The encoding is $52.0 + (0.5 * \text{StcTmpLmt})$ , ranging from 52.0 Tctl to 115.5 Tctl.
15	Reserved.
14:12	Reserved.
11	Reserved.
10:8	Reserved.
7	<b>StcTmpLoSts: STC temperature low status.</b> Read; write-1-to-clear. This bit is set high when the processor exits the STC thermal zone.
6	<b>StcTmpHiSts: STC temperature high status.</b> Read; write-1-to-clear. This bit is set high when the processor enters the STC thermal zone.
5	<b>StcPstateEn: STC P-state enable.</b> Read-write. 1=Place the processor into the STC-active state.
4	Reserved.

3	<b>StcApcTmpLoEn: STC APIC temperature low interrupt enable.</b> Read-write. 1=Enables the generation of an interrupt using [The Thermal Local Vector Table Entry] APIC330 of each core when the processor exits the STC thermal zone. This bit should always be 0.
2	<b>StcApcTmpHiEn: STC APIC temperature high interrupt enable.</b> Read-write. 1=Enables the generation of an interrupt using [The Thermal Local Vector Table Entry] APIC330 of each core when the processor enters the STC thermal zone. This bit should always be 0.
1	<b>StcSbcTmpLoEn: STC special bus cycle temperature low enable.</b> Read-write. 1=Enables the generation of a link special bus cycle (SysMgtCmd = 1101_xxx1b, processor thermal trip point crossed) when the processor exits the STC thermal zone. This bit should always be 0.
0	<b>StcSbcTmpHiEn: STC special bus cycle temperature high enable.</b> Read-write. 1=Enables the generation of a link special bus cycle (SysMgtCmd = 1101_xxx1b, processor thermal trip point crossed) when the processor enters the STC thermal zone. This bit should always be 0.

### F3x6C Data Buffer Count Register

Reset: See field definitions. Updates to this register do not take effect until after a warm reset.

- To ensure deadlock free operation the following minimum buffer allocations are required:  
 $UpRspDBC \geq 1$                        $DnReqDBC \geq 1$                        $UpReqDBC \geq 1$   
 $DnRspDBC \geq 1$
- If the node is directly connected to an IO link with an IOMMU present,  $IsocRspDBC$  must be  $\geq 1$ .
- If  $F0x68[DispRefModeEn]$  is set or any of the  $F0x[E4, C4, A4, 84][IsocEn]$  bits are set:  $IsocRspDBC \geq 1$
- The total number of data buffers allocated in this register and **F3x7C** must satisfy the following equation if  $DatBuf24=0$ :  
 $IsocRspDBC + UpRspDBC + DnReqDBC + UpReqDBC + DnRspDBC + F3x7C[Sri2XbarFreeXreqDBC] + F3x7C[Sri2XbarFreeRspDBC] \leq 16$
- The total number of data buffers allocated in this register and **F3x7C** must satisfy the following equation if  $DatBuf24=1$ :  
 $IsocRspDBC + UpRspDBC + DnReqDBC + UpReqDBC + DnRspDBC + F3x7C[Sri2XbarFreeXreqDBC] + F3x7C[Sri2XbarFreeRspDBC] \leq 24$

See 2.6.4.2.5.1 [Recommended Buffer Count Settings for Single Link Processors] and 2.6.4.2.5.2 [Recommended Buffer Count Settings for Multi-link Processors].

Bits	Description
31	Reserved.
30:28	<b>IsocRspDBC: isochronous response data buffer count.</b> Read-write. Cold reset: 3.
27:19	Reserved.
18:16	<b>UpRspDBC: upstream response data buffer count.</b> Read-write. Cold reset: 2.
15	<b>DatBuf24: data buffer allocation 24.</b> Read-write. Cold reset: 0. 1=24 SRI to XBAR data buffers allocated. 32 SRI to XBAR command buffers allocated. 0=16 SRI to XBAR data buffers allocated. 64 SRI to XBAR command buffers allocated.
14:8	Reserved.
7:6	<b>DnRspDBC: downstream response data buffer count.</b> Read-write. Cold reset: 2.
5:4	<b>DnReqDBC: downstream request data buffer count.</b> Read-write. Cold reset: 1.
3	Reserved.
2:0	<b>UpReqDBC: upstream request data buffer count.</b> Read-write. Cold reset: 2.

### F3x70 SRI to XBAR Command Buffer Count Register

Updates to this register do not take effect until after a warm reset.

- To ensure deadlock free operation the following minimum buffer allocations are required:
 

UpRspCBC >= 1	UpPreqCBC >= 1	DnPreqCBC >= 1
UpReqCBC >= 1	DnReqCBC >= 1	DnRspCBC >= 1
- If F0x68[DispRefModeEn] is set or any of the F0x[E4, C4, A4, 84][IsocEn] bits are set:
 

IsocReqCBC >= 1	IsocRspCBC >= 1
-----------------	-----------------
- If any of the F0x[E4, C4, A4, 84][IsocEn] bits are set 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 F3x7C must satisfy the following equation if F3x6C[DatBuf24]=0:
 
$$\text{IsocRspCBC} + \text{IsocPreqCBC} + \text{IsocReqCBC} + \text{UpRspCBC} + \text{DnPreqCBC} + \text{UpPreqCBC} + \text{DnReqCBC} + \text{DnRspCBC} + \text{UpReqCBC} + \text{F3x7C}[\text{Sri2XbarFreeRspCBC}] + \text{F3x7C}[\text{Sri2XbarFreeXreqCBC}] \leq 64$$
- The total number of SRI to XBAR command buffers allocated in this register and F3x7C must satisfy the following equation if F3x6C[DatBuf24]=1:
 
$$\text{IsocRspCBC} + \text{IsocPreqCBC} + \text{IsocReqCBC} + \text{UpRspCBC} + \text{DnPreqCBC} + \text{UpPreqCBC} + \text{DnReqCBC} + \text{DnRspCBC} + \text{UpReqCBC} + \text{F3x7C}[\text{Sri2XbarFreeRspCBC}] + \text{F3x7C}[\text{Sri2XbarFreeXreqCBC}] \leq 32$$

See 2.6.4.2.5.1 [Recommended Buffer Count Settings for Single Link Processors] and 2.6.4.2.5.2 [Recommended Buffer Count Settings for Multi-link Processors].

Bits	Description
31	Reserved.
30:28	<b>IsocRspCBC: isochronous response command buffer count.</b> Read-write. Cold reset: 6.
27	Reserved.
26:24	<b>IsocPreqCBC: isochronous posted request command buffer count.</b> Read-write. Cold reset: 1.
23	Reserved.
22:20	<b>IsocReqCBC: isochronous request command buffer count.</b> Read-write. Cold reset: 7.
19	Reserved.
18:16	<b>UpRspCBC: upstream response command buffer count.</b> Read-write. Cold reset: 4.
15	Reserved.
14:12	<b>DnPreqCBC: downstream posted request command buffer count.</b> Read-write. Cold reset: 4.
11	Reserved.
10:8	<b>UpPreqCBC: upstream posted request command buffer count.</b> Read-write. Cold reset: 4.
7:6	<b>DnRspCBC: downstream response command buffer count.</b> Read-write. Cold reset: 2.
5:4	<b>DnReqCBC: downstream request command buffer count.</b> Read-write. Cold reset: 2.
3	Reserved.
2:0	<b>UpReqCBC: upstream request command buffer count.</b> Read-write. Cold reset: 4.

### F3x74 XBAR to SRI Command Buffer Count Register

Updates to this register do not take effect until after a warm reset.



- To ensure deadlock free operation in a multi-node system the following minimum buffer allocations are required:
 

ProbeCBC $\geq$ 2	DnPreqCBC $\geq$ 1	UpPreqCBC $\geq$ 1
DnReqCBC $\geq$ 1	UpReqCBC $\geq$ 1	
- To ensure deadlock free operation in a single node system the following minimum buffer allocations are required:
 

ProbeCBC $\geq$ 2	UpReqCBC $\geq$ 1	UpPreqCBC $\geq$ 1
-------------------	-------------------	--------------------
- To ensure deadlock free operation ProbeCBC must be less than or equal to 8.
- If F0x68[DispRefModeEn] is set or the node is directly connected to an IO link with an IOMMU present: IsocReqCBC  $\geq$  1
- If any of the F0x[E4, C4, A4, 84][IsocEn] bits are set: IsocReqCBC  $\geq$  1
- If any of the F0x[E4, C4, A4, 84][IsocEn] bits are set and isochronous posted requests may be generated by the system: IsocPreqCBC  $\geq$  1
- If F0x68[DispRefModeEn] is set or if any of the F0x[E4, C4, A4, 84][IsocEn] bits are set and F3x158[LnkToXcsDRToken]  $>$ 0:
 

IsocPreqCBC $\geq$ 1	IsocReqCBC $\geq$ 1	DRReqCBC $\geq$ 1
----------------------	---------------------	-------------------
- The total number of XBAR to SRI command buffers allocated in this register and F3x7C must satisfy the following equation if the processor includes a L3 cache:
 
$$\text{DRReqCBC} + \text{IsocPreqCBC} + \text{IsocReqCBC} + \text{DnPreqCBC} + \text{UpPreqCBC} + \text{DnReqCBC} + \text{UpReqCBC} + \text{F3x7C}[\text{Xbar2SriFreeListCBC}] + \text{F3x1A0}[\text{L3ToSriReqCBC}] \leq 32$$
- The total number of XBAR to SRI command buffers allocated in this register and F3x7C must satisfy the following equation if the processor does not include a L3 cache:
 
$$\text{DRReqCBC} + \text{IsocPreqCBC} + \text{IsocReqCBC} + \text{DnPreqCBC} + \text{UpPreqCBC} + \text{DnReqCBC} + \text{UpReqCBC} + \text{F3x7C}[\text{Xbar2SriFreeListCBC}] + (\text{F3x1A0}[\text{CpuCmdBufCnt}] * (\text{F3xE8}[\text{CmpCap}] + 1)) \leq 32$$

See 2.6.4.2.5.1 [Recommended Buffer Count Settings for Single Link Processors] and 2.6.4.2.5.2 [Recommended Buffer Count Settings for Multi-link Processors].

Bits	Description
31:28	<b>DRReqCBC: display refresh request command buffer count.</b> Read-write. Cold reset: 0.
27	Reserved.
26:24	<b>IsocPreqCBC: isochronous posted request command buffer count.</b> Read-write. Cold reset: 0.
23:20	<b>IsocReqCBC: isochronous request command buffer count.</b> Read-write. Cold reset: 4.
19:16	<b>ProbeCBC: probe command buffer count.</b> Read-write. Cold reset: 8.
15	Reserved.
14:12	<b>DnPreqCBC: downstream posted request command buffer count.</b> Read-write. Cold reset: 1.
11	Reserved.
10:8	<b>UpPreqCBC: upstream posted request command buffer count.</b> Read-write. Cold reset: 1.
7	Reserved.
6:4	<b>DnReqCBC: downstream request command buffer count.</b> Read-write. Cold reset: 1.
3	Reserved.
2:0	<b>UpReqCBC: upstream request command buffer count.</b> Read-write. Cold reset: 1.

### F3x78 MCT to XBAR Buffer Count Register



Updates to this register do not take effect until after a warm reset.

- To ensure deadlock free operation the following minimum buffer allocations are required:  
 $\text{ProbeCBC} \geq 1$                        $\text{RspCBC} \geq 1$                        $\text{RspDBC} \geq 2$   
 Revision C and earlier revisions:  $\text{RspDBC} \geq \text{F2x11C}[\text{MctPrefReqLimit}] + 1$   
 Revision D and later:  $\text{RspDBC} \geq \text{F2x11C}[\text{MctPrefReqLimit}] + 2$
- To ensure deadlock free operation when online spare is enabled ( $\text{F2x}[1, 0][5\text{C}:40][\text{Spare}] = 1$ ) the following minimum buffer allocation is required:  
 $\text{RspCBC} \geq 13$
- The total number of command buffers allocated in this register must satisfy the following equation:  
 $\text{ProbeCBC} + \text{RspCBC} \leq 32$

Bits	Description
31:22	Reserved.
21:16	<b>RspDBC: response data buffer count.</b> Read-write. Cold reset: 32. Valid values are greater than or equal to 2 and less than or equal to 32.
15:13	Reserved.
12:8	<b>ProbeCBC: probe command buffer count.</b> Read-write. Cold reset: Ch.
7:5	Reserved.
4:0	<b>RspCBC: response command buffer count.</b> Read-write. Cold reset: 14h.

### **F3x7C Free List Buffer Count Register**

Updates to this register do not take effect until after a warm reset.

To ensure deadlock free operation the following minimum buffer allocations are required:

- If  $\text{Sri2XbarFreeRspCBC} = 0$ :  $\text{Sri2XbarFreeXreqCBC} > 2$
- If  $\text{Sri2XbarFreeRspCBC} \neq 0$ :  $\text{Sri2XbarFreeRspCBC} > 2$
- If  $\text{Sri2XbarFreeRspDBC} = 0$ :  $\text{Sri2XbarFreeXreqDBC} > 2$
- If  $\text{Sri2XbarFreeRspDBC} \neq 0$ :  $\text{Sri2XbarFreeRspDBC} > 2$
- $\text{Xbar2SriFreeListCBC} \geq 2$

See 2.6.4.2.5.1 [Recommended Buffer Count Settings for Single Link Processors] and 2.6.4.2.5.2 [Recommended Buffer Count Settings for Multi-link Processors].

Bits	Description
31	Reserved.
30:28	<b>Xbar2SriFreeListCBInc: XBAR to SRI free list command buffer increment.</b> Read-write. Cold reset: 0. This is used to add buffers to the free list pool if they are reclaimed from hard allocated entries without having to go through warm reset.
27:23	Reserved.
22:20	<b>Sri2XbarFreeRspDBC: SRI to XBAR free response data buffer count.</b> Read-write. Cold reset: 3.
19:16	<b>Sri2XbarFreeXreqDBC: SRI to XBAR free request and posted request data buffer count.</b> Read-write. Cold reset: 3. When $\text{Sri2XbarFreeRspDBC} = 0$ , 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. Cold reset: 15.
11:8	<b>Sri2XbarFreeXreqCBC: SRI to XBAR free request and posted request command buffer count.</b> Read-write. Cold reset: 15. When Sri2XbarFreeRspCBC=0h, these buffers are shared between requests, responses and posted requests and the number of buffers allocated is two times the value of this field.
7:5	Reserved.
4:0	<b>Xbar2SriFreeListCBC: XBAR to SRI free list command buffer count.</b> Read-write. Cold reset: varies based on the state of F3xE8[CmpCap] and whether the processor includes L3 cache: <ul style="list-style-type: none"> <li>• 1-core without L3 cache is 22.</li> <li>• 2-core or any processor with L3 cache is 20.</li> <li>• 3-core without L3 cache is 18.</li> <li>• 4-core without L3 cache is 16.</li> </ul>

### **F3x[84:80] ACPI Power State Control Registers**

Reset: 0000 0000h. 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 table below for the associated ACPI state and SMAF code for each of the 8 registers. Some ACPI states and associated SMAF codes may not be supported in certain conditions. Refer to section 2.4 [Power Management] for information on 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.

Note: in multi-node systems, these registers should be programmed identically in all nodes.

**Table 100: ACPI Power State Control Register SMAF Settings**

ACPI State	SMAF Code	Description/Initiation	Register/Setting
C2	000b	Initiated by a processor access to the ACPI-defined P_LVL2 register.	F3x80[7:0]: 81h
C3, C1E, or Link init	001b	Initiated by a processor access to the ACPI-defined P_LVL3 register or in response to a write to the Link Frequency Change and Resize LDTSTOP_L Command register in the IO hub. LDTSTOP_L is expected to be asserted while in this state. C3 and C1e support varies by revision. See Table 2 and 2.4.3.3 [C1 Enhanced State (C1E)]. See 2.7.7 [LDTSTOP Requirements].	F3x80[15:8]: If the link is at a Gen1 frequency and the chipset does not support a 10us minimum LDTSTOP assertion time: If ASB2 && F3xA0[SviHighFreqSel]=1: F6h. Else: 87h. Else: Revision B: A6h. Revision C2: E6h. Revision C3: If F3xA0[SviHighFreqSel]=1: F6h. Else: E6h. Revision D1 C32 and G34 processors: E6h. Revision E: E6h.

**Table 100: ACPI Power State Control Register SMAF Settings**

FIDVID change	010b	This is used for NB P-state changes. See 2.4.2.2 [NB P-states].	F3x80[23:16]: Revision DA-C in S1g3 and revision C3: 0Bh. All other revisions: 00h.
S1	011b	Initiated by a processor access to the ACPI-defined PM1_CNTa register.	F3x80[31:24]: E6h
S3	100b	Initiated by a processor access to the ACPI-defined PM1_CNTa register.	F3x84[7:0]: E6h
Throttling	101b	Occurs based upon SMC hardware-initiated throttling. AMD recommends using PROCHOT_L for thermal throttling and not implementing stop clock based throttling. Due to system interactions, throttling settings in the chipset must be configured such that STPCLK_L is deasserted for a sufficient amount of time to make forward progress. The amount of time varies with the number of cores and Clk-Divisor settings. Deeper ClkDivisor settings or greater numbers of cores require additional STPCLK_L deassertion time.	F3x84[15:8]: 41h
S4/S5	110b	Initiated by a processor access to the ACPI-defined PM1_CNTa register.	F3x84[23:16]:E6h
C1	111b	Initiated when a HLT 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.	F3x84[31:24]: Revision B with L2 and data cache scrubbing disabled: A0h Revision B with L2 or data cache scrubbing enabled and revision C and later with cache flush on halt disabled: 80h. Revision C and later with cache flush on halt enabled: 01h. See 2.6.6 [Memory Scrubbers] and 2.4.3.4 [Cache Flush On Halt].
1. See 2.6.4.2.5 [Display Refresh And IFCM].			

Bits	Description																				
31:8	See above.																				
7:5	<p><b>ClkDivisor: 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[68:64][CpuFid]) of the current P-state specified by MSRC001_0063[CurPstate].</li> </ul> <p>If MSRC001_00[68:64][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. This field is encoded as follows:</p> <table border="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Divisor</u></th> <th><u>Bits</u></th> <th><u>Divisor</u></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>Divide-by 128.</td> </tr> <tr> <td>010b</td> <td>Divide-by 4.</td> <td>110b</td> <td>Divide-by 512.</td> </tr> <tr> <td>011b</td> <td>Divide-by 8.</td> <td>111b</td> <td>Turn off clocks.</td> </tr> </tbody> </table> <p>See 2.6.6 [Memory Scrubbers].</p>	<u>Bits</u>	<u>Divisor</u>	<u>Bits</u>	<u>Divisor</u>	000b	Divide-by 1.	100b	Divide-by 16.	001b	Divide-by 2.	101b	Divide-by 128.	010b	Divide-by 4.	110b	Divide-by 512.	011b	Divide-by 8.	111b	Turn off clocks.
<u>Bits</u>	<u>Divisor</u>	<u>Bits</u>	<u>Divisor</u>																		
000b	Divide-by 1.	100b	Divide-by 16.																		
001b	Divide-by 2.	101b	Divide-by 128.																		
010b	Divide-by 4.	110b	Divide-by 512.																		
011b	Divide-by 8.	111b	Turn off clocks.																		

4	<b>AltVidEn: alternate VID enable.</b> Read-write. 1=The alternate VID is driven while in the low-power state. If <a href="#">F3xA0[PviMode]=1</a> , then the normal VID is applied; if <a href="#">F3xA0[PviMode]=0</a> , then <a href="#">F3xDC[AltVid]</a> is applied to the core power plane and the normal VID is applied to VDDNB. See <a href="#">2.4.1.5 [Alternative Voltage (Altvid)]</a> .
3	Reserved.
2	<b>NbGateEn: Northbridge gate enable.</b> Read-write. 1=MEMCLK is tristated when DRAM is in self-refresh mode and LDTSTOP_L is asserted while in the low-power state. NbLowPwrEn is required to be set if this bit is set.
1	<b>NbLowPwrEn: Northbridge low-power enable.</b> Read-write. 1=The NB clock is ramped down to the divisor specified by <a href="#">[The Clock Power/Timing Control 0 Register] F3xD4[NbClkDiv]</a> when LDTSTOP_L is asserted while in the low-power state.
0	<b>CpuPrbEn: CPU direct probe enable.</b> Read-write. Specifies how probes are handled while in the low-power state. For revision C and later, this bit also specifies functionality of the timer used for cache flushing during halt (see <a href="#">F3xDC[CacheFlushOnHaltTmr]</a> ). 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 <a href="#">[The Clock Power/Timing Control 0 Register] F3xD4[ClkRampHystSel]</a> , 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: <ul style="list-style-type: none"> <li>• ClkDivisor specifies a divide-by 1, 2, 4, 8, or 16 and NB COF &lt;= 3.2 GHz</li> <li>• ClkDivisor specifies a divide-by 1, 2, 4, or 8 and NB COF &gt;= 3.4 GHz</li> </ul> This bit should be set if probes are expected to occur while in the low-power state associated with the SMAF.

### **F3x[8C:88] NB Configuration High, Low Registers**

These addresses form a duplicated access space for [MSRC001\\_001F](#). See [MSRC001\\_001F\[31:0\]](#) for F3x88 and [MSRC001\\_001F\[63:32\]](#) F3x8C.

### **F3x90 GART Aperture Control Register**

Reset: 0000 0000h. Note: GART apertures or translations above 1 terabyte are not supported.

Bits	Description
31:7	Reserved.
6	<b>DisGartTblWlkPrb: disable GART table walk probes.</b> Read-write. 1=Disables generation of probes for GART table walks. This bit may be set to improve performance in cases where the GART table entries are in address space which is marked uncacheable in processor MTRRs or page tables.
5	<b>DisGartIo: disable GART IO accesses.</b> Read-write. 1=Disables requests from IO devices from accessing the GART.
4	<b>DisGartCpu: disable GART CPU accesses.</b> Read-write. 1=Disables requests from CPUs from accessing the GART.

3:1	<b>GartSize: GART size.</b> Read-write. Specifies the size of address space allocated to the GART. 000b = 32 Mbytes 001b = 64 Mbytes 010b = 128 Mbytes 011b = 256 Mbytes 100b = 512 Mbytes 101b = 1 Gbyte 110b = 2 Gbytes 111b = Reserved
0	<b>GartEn: GART enable.</b> Read-write. 1=Enables GART address translation for accesses falling within the GART aperture. <b>F3x94</b> [GartAperBaseAddr] and other related registers should be initialized before GartEn is set.

### F3x94 GART Aperture Base Register

Bits	Description
31:15	Reserved.
14:0	<b>GartAperBaseAddr[39:25]: GART aperture base address bits[39:25].</b> Read-write. Reset: X. Specifies the base address of the GART aperture range. Based on <b>F3x90</b> [GartSize], some of the LSB address bits are assumed to be 0 (e.g., if the GART is 1 Gbyte, then only GartAperBaseAddr[39:30] is meaningful). This field along with <b>F3x90</b> [GartSize] specifies the GART aperture address range. BIOS can place the GART aperture below the 4-gigabyte level in address space in order to support legacy operating systems and legacy AGP cards (that do not support 64-bit address space). Note: GART apertures above 1 terabyte are not supported.

### F3x98 GART Table Base Register

Reset: xxxx xxx0h.

Bits	Description												
31:4	<b>GartTblBaseAddr[39:12]: GART table base address bits[39:12].</b> Read-write. Specifies the base address of the table of GART page table entries (PTEs) used in GART address translation. Accesses to the GART aperture address range specified by <b>F3x90</b> and <b>F3x94</b> , address GA[39:0], are translated to the physical address specified by the corresponding GART PTE. Each PTE is 32-bits wide. The first PTE corresponds to the first 4 KByte page of the GART aperture, and so on. PTEs are defined as follows: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>PTE bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>31:12</td> <td>Physical address bits[31:12].</td> </tr> <tr> <td>11:4</td> <td>Physical address bits[39:32].</td> </tr> <tr> <td>3:2</td> <td>Reserved.</td> </tr> <tr> <td>1</td> <td>Coherent: 1=Probes are required for accesses to the range.</td> </tr> <tr> <td>0</td> <td>Valid: 1=Entry is valid.</td> </tr> </tbody> </table> <p>GART translations to addresses above 1 terabyte are not supported. The page table is required to reside within DRAM address ranges. Also, the page tables are expected to translate to DRAM address ranges only; translations to MMIO ranges result in undefined behavior.</p>	PTE bits	Description	31:12	Physical address bits[31:12].	11:4	Physical address bits[39:32].	3:2	Reserved.	1	Coherent: 1=Probes are required for accesses to the range.	0	Valid: 1=Entry is valid.
PTE bits	Description												
31:12	Physical address bits[31:12].												
11:4	Physical address bits[39:32].												
3:2	Reserved.												
1	Coherent: 1=Probes are required for accesses to the range.												
0	Valid: 1=Entry is valid.												
3:0	Reserved.												

### F3x9C GART Cache Control Register

Reset: 0000 0000h.

Bits	Description
31:2	Reserved.
1	<b>GartPteErr: GART PTE error.</b> Read; set-by-hardware; write-1-to-clear. 1=An invalid PTE was encountered during a table walk.
0	<b>InvGart: invalidate GART.</b> Read; write-1-only. Setting this bit causes the GART cache to be invalidated. This bit is cleared by hardware when the invalidation is complete.

### F3xA0 Power Control Miscellaneous Register

Reset: 0000 0000h except bit 9.

Bits	Description																
31	<b>CofVidProg: COF and VID of P-states programmed.</b> Read-only. 1=Out of cold reset, the VID and FID values of the P-state register specified by <a href="#">MSRC001_0071</a> [StartupPstate] 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). This affects <a href="#">F3xD4</a> [NbFidEn]. 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="#">F3xA0</a> [CofVidProg]. BIOS must transition the processor to a valid P-state out of cold reset when <a href="#">F3xA0</a> [CofVidProg]=0. See <a href="#">2.4.2.6</a> [BIOS Requirements for P-State Initialization and Transitions].																
30	Reserved.																
29	<b>SlamVidMode: slam voltage ID mode.</b> Read-write. This specifies the voltage transition type when changing P-state. 1=The voltage is slammed. 0=The voltage is stepped. This bit is normally set if the regulator includes built-in output voltage slew rate control. It is required to be programmed to the same state in all nodes. See <a href="#">2.4.1.8</a> [Hardware-Initiated Voltage Transitions]. BIOS should set this bit to the inverse of <a href="#">F3xA0</a> [PviMode].																
28	<b>NbPstateForce: NB P-state force on next LDTSTOP assertion.</b> Read-write. 1=When there is a pending NB P-state change (after the FIDVID broadcast), the NB P-state is changed on the next LDTSTOP assertion regardless of whether the SMAF code for the assertion matches <a href="#">[The ACPI Power State Control Registers] F3x[84:80]</a> in which NbCofChg is set. BIOS should set this bit for revision DA-C and C3.																
27:16	<b>PstateId: P-state identifier.</b> Read-only. This field specifies the P-state ID associated with the product.																
15	Reserved.																
14	<b>BpPinsTriEn: breakpoint pins tristate enable.</b> Revision D0 and earlier: Reserved. Revision D1 and later: Read-write. 1=Tristate the BP[5:0] pins. If IdleExitEn=1, then BP[5] is not tristated regardless of BpPinsTriEn. BIOS should program this bit to 1 for G34 and C32 processors.																
13:11	<b>PlLockTime: PLL synchronization lock time.</b> Read-write. 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. These bits are encoded as follows: <table border="0" style="margin-left: 20px;"> <tr> <td>000b</td> <td>1 microsecond.</td> <td>100b</td> <td>8 microseconds.</td> </tr> <tr> <td>001b</td> <td>2 microseconds.</td> <td>101b</td> <td>16 microseconds.</td> </tr> <tr> <td>010b</td> <td>3 microseconds.</td> <td>110b</td> <td>Reserved.</td> </tr> <tr> <td>011b</td> <td>4 microseconds.</td> <td>111b</td> <td>Reserved.</td> </tr> </table> For revision B, BIOS should set this field to 101b. For revision C and later, BIOS should set this field to 001b.	000b	1 microsecond.	100b	8 microseconds.	001b	2 microseconds.	101b	16 microseconds.	010b	3 microseconds.	110b	Reserved.	011b	4 microseconds.	111b	Reserved.
000b	1 microsecond.	100b	8 microseconds.														
001b	2 microseconds.	101b	16 microseconds.														
010b	3 microseconds.	110b	Reserved.														
011b	4 microseconds.	111b	Reserved.														



10	<b>IdleExitEn.</b> Revision D0 and earlier: Reserved. Revision D1 and later: Read-write. 1=Enable BP[5] to function as the IDLE_EXIT_L pin. See BpPinsTriEn and <a href="#">2.4.3.3.5 [BIOS Requirements to Initialize Message Triggered C1E]</a> . This bit is only supported for revision D1 and later G34 and C32 processors.
9	<b>SviHighFreqSel: SVI high frequency select.</b> Revision C2 and earlier and revision D and later: Reserved. Revision C3: Read-write. Cold reset: 0. 0=400 KHz. 1=3.4 MHz. Writes to this field take effect at the next SVI command boundary. If 3.4 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.
8	<b>PviMode: parallel VID interface mode.</b> Read-only. 1=The parallel VID interface is selected; single- or dual-plane operation. 0=The serial VID interface is selected; dual-plane operation. See. <a href="#">2.4.1.1 [VID Pins And Interface Selection]</a> .
7	<b>PsiVidEn: PSI_L VID enable.</b> Read-write. 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 <a href="#">2.4.1.4 [PSI_L]</a> .
6:0	<b>PsiVid: PSI_L VID threshold.</b> Read-write. When enabled by PsiVidEn, this field specifies the threshold value of VID code generated by the processor, which in turn determines the state of PSI_L. When the VID code generated by the processor is less than PsiVid (i.e., the VID code is specifying a higher voltage level than the PsiVid-specified voltage level), then PSI_L is high; when the VID code is greater than or equal to PsiVid, PSI_L is driven low. See <a href="#">2.4.1.4 [PSI_L]</a> .

### F3xA4 Reported Temperature Control Register

The processor measures temperature to 1/2-degree C resolution. However, temperature is reported through Tctl with 1/8th-degree resolution. The translation to finer resolution is accomplished using slew rate controls in this register. These specify how quickly Tctl steps to the measured temperature in 1/8th-degree steps. Separate controls are provided for measured temperatures that are higher and 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 flops to the other side of Tctl, the step timer resets. 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.

Bits	Description
31:21	<b>CurTmp: current temperature.</b> Read-only. Reset: X. Provides the current control temperature, Tctl (after the slew-rate controls have been applied). This is encoded as value = 1/8th degree * Tctl, ranging from 0 to 255.875 degrees. See <a href="#">2.10.1 [The Tctl Temperature Scale]</a> and CurTmpSel. If <a href="#">F3xE8[MultiNodeCpu]=1</a> , this field should be accessed on internal node 0 ( <a href="#">F3xE8[IntNodeNum]=00b</a> ) to obtain temperature information for the processor. Reading this field on internal node 1 provides only the temperature for internal node 1. If CurTmpSel=11b, programming this field on internal node 1 does not affect CurTmp or temperature-driven logic on internal node 0.
20:18	Reserved.



17:16	<b>CurTmpSel.</b> Current temperature select. Read-write. Reset: 00. These bits may be used for diagnostic software. This bits are encoded as: 00b = CurTmp provides the read-only Tctl value. 01b = Undefined. 10b = Undefined. 11b = CurTmp is a read-write register that specifies a value, used to create Tctl. The two LSBs are read-only zero.										
15:13	Reserved.										
12:8	<b>PerStepTimeDn[4:0]: per 1/8th degree step time down.</b> Read-write. Cold reset: 18h (1 second). This specifies the time per 1/8-degree step of Tctl when the measured temperature is less than the Tctl. It is encoded the same as PerStepTimeUp. For S1g3, G34, C32, and revision C3 and E processors, BIOS should program this to 0Fh.										
7	<b>TmpSlewDnEn: temperature slew downward enable.</b> Read-write. Cold reset: 0b. 1=Slew rate controls in the downward direction are enabled. 0=Downward slewing disabled; if the measured temperature is detected to be less than Tctl then Tctl is updated to match the measured temperature. BIOS should set this bit to 1.										
6:5	<b>TmpMaxDiffUp: temperature maximum difference up.</b> Read-write. Cold reset: 000b. This specifies the maximum difference between Tctl and the measured temperature, when the measured value is greater than Tctl (i.e., when the temperature has risen). If this difference exceeds the specified value, Tctl jumps to the measured temperature value. This field is encoded as follows: 00b = Upward slewing disabled; if the measured temperature is detected to be greater than Tctl then Tctl is updated to match the measured temperature. 01b = Tctl is held to less than or equal to measured temperature minus 1.0 degrees C. 10b = Tctl is held to less than or equal to measured temperature minus 3.0 degrees C. 11b = Tctl is held to less than or equal to measured temperature minus 9.0 degrees C. For S1g3, G34, C32, and revision C3 and E processors, BIOS should program this to 11b.										
4:0	<b>PerStepTimeUp[4:0]: per 1/8th degree step time up.</b> Read-write. Cold reset: 00h. This specifies the time per 1/8-degree step of Tctl when the measured temperature is greater than the reported temperature. It is encoded as follows: <table border="1"> <thead> <tr> <th>Bits[4:3]</th> <th>Step Time</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>(Bits[2:0] + 1) * 1 millisecond, ranging from 1 to 8 milliseconds.</td> </tr> <tr> <td>01b</td> <td>(Bits[2:0] + 1) * 10 millisecond, ranging from 10 to 80 milliseconds.</td> </tr> <tr> <td>10b</td> <td>(Bits[2:0] + 1) * 100 millisecond, ranging from 100 to 800 milliseconds.</td> </tr> <tr> <td>11b</td> <td>(Bits[2:0] + 1) * 1 second, ranging from 1 to 8 seconds.</td> </tr> </tbody> </table> For S1g3, G34, C32, and revision C3 and E processors, BIOS should program this to 0Fh.	Bits[4:3]	Step Time	00b	(Bits[2:0] + 1) * 1 millisecond, ranging from 1 to 8 milliseconds.	01b	(Bits[2:0] + 1) * 10 millisecond, ranging from 10 to 80 milliseconds.	10b	(Bits[2:0] + 1) * 100 millisecond, ranging from 100 to 800 milliseconds.	11b	(Bits[2:0] + 1) * 1 second, ranging from 1 to 8 seconds.
Bits[4:3]	Step Time										
00b	(Bits[2:0] + 1) * 1 millisecond, ranging from 1 to 8 milliseconds.										
01b	(Bits[2:0] + 1) * 10 millisecond, ranging from 10 to 80 milliseconds.										
10b	(Bits[2:0] + 1) * 100 millisecond, ranging from 100 to 800 milliseconds.										
11b	(Bits[2:0] + 1) * 1 second, ranging from 1 to 8 seconds.										

### F3xA8 Popup P-state Register

Reset: 0000 0000h except bits 31:29.

Bits	Description
31:29	<b>PopDownPstate.</b> Revision D1 and earlier: Reserved. Revision E: Read-write. Reset: F3xDC[PstateMaxVal]. Specifies the pop-down P-state number. This field uses hardware P-state numbering.
28	Reserved.

27	<b>CacheFlushPopDownEn.</b> Revision D1 and earlier: Reserved. Revision E: Read-write. 1=Cores automatically transition to the pop-down P-state after flushing caches in a non-C0 C-state. 0=Cores do not automatically transition to the pop-down P-state in a non-C0 C-state. BIOS should set this bit if $F4x15C[NumBoostStates] \neq 0$ . See 2.4.2.1.1 [Core Performance Boost (CPB)].
26:0	Reserved.

### F3xB0 On-Line Spare Control Register

See 2.8.12 [On-Line Spare].

Bits	Description
31:28	<b>LvtOffset: local vector table offset.</b> Reset: 0000b. This 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]). This offset applies to SwapDoneInt and EccErrInt interrupts.
27:24	<b>EccErrCnt: ECC error count.</b> Read-write (modified by EccErrCntWrEn). Reset: 0. This field returns the number of ECC errors for the chip select selected by the EccErrCntDramCs, and EccErrCntChan. This field can be written by software to clear the count. This field returns Fh if 15 or more correctable ECC errors have occurred.
23	<b>EccErrCntWrEn: ECC error counter write enable.</b> Read-write. Reset: 0. 1=Enable writes to the EccErrCnt field.
22:21	Reserved.
20	<b>EccErrCntChan: ECC error counter channel.</b> Read-write. Reset: 0. Specifies the DCT for which ECC error count information is returned in the EccErrCnt field. 0=DCT0. 1=DCT1.
19:16	<b>EccErrCntDramCs[3:0]: ECC error counter DRAM chip select.</b> Read-write. Reset: 0. This field specifies the DRAM chip select (as specified in [The DRAM CS Base Address Registers] F2x[1, 0][5C:40]) for which ECC error count information is returned in the EccErrCnt field. Depending only the production, this field is interpreted as follows: <ul style="list-style-type: none"> <li>• DDR products: EccErrCntDramCs[3] does not matter and EccErrCntDramCs[2:0] specifies the error count for the address range of one of the eight chip select specified by [The DRAM CS Base Address Registers] F2x[1, 0][5C:40].</li> </ul>
15:14	<b>EccErrInt: ECC error interrupt type.</b> Read-write. Reset: 0. This field specifies the type of interrupt generated when the EccErrCnt field for any chip select and channel transitions to 1111b. <ul style="list-style-type: none"> <li>00b = No Interrupt.</li> <li>01b = APIC based interrupt (see LvtOffset) to all cores.</li> <li>10b = SMI trigger event (always routed to CpuCoreNum 0, as defined in section 2.9.2 [CPU Cores and Downcoring]); see 2.13.2.3 [SMI Sources And Delivery].</li> <li>11b = Reserved.</li> </ul>
13:12	<b>SwapDoneInt: swap done interrupt type.</b> Read-write. Reset: 0. This field specifies the type of interrupt generated when a swap is complete. <ul style="list-style-type: none"> <li>00b = No Interrupt.</li> <li>01b = APIC based interrupt (see LvtOffset) to all cores.</li> <li>10b = SMI trigger event (always routed to CpuCoreNum 0, as defined in section 2.9.2 [CPU Cores and Downcoring]); see 2.13.2.3 [SMI Sources And Delivery].</li> <li>11b = Reserved.</li> </ul>
11	Reserved.
10:8	<b>BadDramCs1: bad DRAM chip select DCT1.</b> Read-write. Cold reset: 0. See BadDramCs0 below.
7	Reserved.

6:4	<b>BadDramCs0: bad DRAM chip select DCT0.</b> Read-write. Cold reset: 0. This field is programmed with the DRAM chip select to be replaced when SwapEn is set. This field cannot be written when SwapDone is set. BadDramCs0 applies to DCT0 and BadDramCs1 applies to DCT1. BadDramCs0 and BadDramCs1 must be programmed to the same value if the DCTs are operating in ganged mode ( $F2x110[DctGangEn]=1$ ).
3	<b>SwapDone1: swap done DCT1.</b> Read-write; set-by-hardware. Cold reset: 0. See SwapDone0 below.
2	<b>SwapEn1: swap enable DCT1.</b> Read; write-1-only. Reset: 0. See SwapEn0 below.
1	<b>SwapDone0: swap done DCT0.</b> Read-write; set-by-hardware. Cold reset: 0. 1=The hardware has completed copying the data to the spare rank. This bit can also be set by BIOS to immediately enable the swap to the spare rank after suspend to RAM. Once this bit is set it cannot be cleared by software. This bit cannot be set by software if DRAM is enabled $F2x110[DramEnable]$ . SwapDone0 applies to DCT0 and SwapDone1 applies to DCT1.
0	<b>SwapEn0: swap enable DCT0.</b> Read; write-1-only. Reset: 0. Setting this bit causes the hardware to copy the contents of the DRAM chip select identified by BadDramCs to the spare rank. The DRAM scrubber ( $F3x5C$ ) must be enabled with a scrub address range that encompasses the address of the bad chip select for the swap to occur. The scrub rate is accelerated automatically by hardware until the copy completes, at which point the scrub rate returns to normal. During the copy, DRAM accesses (including accesses to the bad CS) proceed normally. Once this bit is set, it cannot be cleared by software. SwapEn0 applies to DCT0 and SwapEn1 applies to DCT1. SwapEn0 and SwapEn1 must be programmed to the same value if the DCTs are operating in ganged mode ( $F2x110[DctGangEn]=1$ ).

### F3xC4 SBI P-state Limit Register

Revision C and earlier: Reserved. Revision D and later: Reset: 0000 0000h. This register is used by the APLM interface. See 2.12.3 [Sideband Interface (SBI)].  $F3xC4$  should only be programmed on internal node 0 ( $F3xE8[IntNodeNum]=00b$ ) if  $F3xE8[MultiNodeCpu]=1$ . This register specifies a P-state limit for all cores in the processor. See 2.4.2.3 [P-state Limits].

Bits	Description
31:11	Reserved.
10:8	<b>PstateLimit: P-state limit select.</b> Read-write from APLM interface; read-only from configuration space. Specifies the P-state limit of all cores when activated by $F3xC4[PstateLimitEn]$ . This field uses hardware P-state numbering and is not changed on a write if the value written is greater than $MSRC001_0061[PstateMaxVal] + F4x15C[NumBoostStates]$ . See 2.4.2.1.2.2 [Hardware P-state Numbering].
7:1	Reserved.
0	<b>PstateLimitEn: P-state limit enable.</b> Read-write from APLM interface; read-only from configuration space. 1=The P-state of all cores is limited to $F3xC4[PstateLimit]$ . 0=The P-state of the cores is not limited by $F3xC4[PstateLimit]$ .

### F3xD4 Clock Power/Timing Control 0 Register

Reset: see field definitions.

Bits	Description
31	<b>NbClkDivApplyAll.</b> Read-write. Cold reset: 0. See NbClkDiv. BIOS should program this bit to 1.

30:28	<p><b>NbClkDiv: NB clock divisor.</b> Read-write. Cold reset: value varies by product. Specifies the NB CLK divisor associated with <a href="#">[The ACPI Power State Control Registers] F3x[84:80][NbLowPwrEn]</a>. This divisor is applied while LDTSTOP is asserted if the corresponding core CLK divisor, <a href="#">F3x[84:80][ClkDivisor]</a>, is set to “turn off clocks” or if <a href="#">NBClkDivApplyAll=1</a>; otherwise, the divisor specified by <a href="#">F3x[84:80][ClkDivisor]</a> is applied. This divisor is relative to the current NB FID frequency, or:</p> <ul style="list-style-type: none"> <li>• 200 MHz * (4 + <a href="#">F3xD4[NbFid]</a>).</li> </ul> <p>If <a href="#">MSRC001_00[68:64][NbDid]</a> of the current 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="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Divisor</u></th> <th><u>Bits</u></th> <th><u>Divisor</u></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<sup>1</sup></td> <td>Divide-by 128.</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> <p>1. This setting is only supported for revision DA-C2 and single link C3 processors and only if the link is running in Gen3 mode.</p> <p>For revision DA-C2 in S1g3 package, if the link is running in Gen3 mode, BIOS should program this field to 101b. For all other cases, BIOS should program this field to 100b.</p>	<u>Bits</u>	<u>Divisor</u>	<u>Bits</u>	<u>Divisor</u>	000b	Divide-by 1.	100b	Divide-by 16.	001b	Divide-by 2.	101b <sup>1</sup>	Divide-by 128.	010b	Divide-by 4.	110b	Reserved.	011b	Divide-by 8.	111b	Reserved.																				
<u>Bits</u>	<u>Divisor</u>	<u>Bits</u>	<u>Divisor</u>																																						
000b	Divide-by 1.	100b	Divide-by 16.																																						
001b	Divide-by 2.	101b <sup>1</sup>	Divide-by 128.																																						
010b	Divide-by 4.	110b	Reserved.																																						
011b	Divide-by 8.	111b	Reserved.																																						
27:24	<p><b>PowerStepUp.</b> Read-write. Cold reset: 0000b. This specifies the rate at which blocks of core 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 12 steps in this transition of each core and about 5 steps for the NB for the PowerStepDown and PowerStepUp transitions. The total transition time for a single core is about 12 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="0"> <thead> <tr> <th><u>Bits</u></th> <th><u>Time</u></th> <th><u>Bits</u></th> <th><u>Time</u></th> <th><u>Bits</u></th> <th><u>Time</u></th> <th><u>Bits</u></th> <th><u>Time</u></th> </tr> </thead> <tbody> <tr> <td>0000b</td> <td>400ns.</td> <td>0100b</td> <td>90ns.</td> <td>1000b</td> <td>50ns.</td> <td>1100b</td> <td>30ns.</td> </tr> <tr> <td>0001b</td> <td>300ns.</td> <td>0101b</td> <td>80ns.</td> <td>1001b</td> <td>45ns.</td> <td>1101b</td> <td>25ns.</td> </tr> <tr> <td>0010b</td> <td>200ns.</td> <td>0110b</td> <td>70ns.</td> <td>1010b</td> <td>40ns.</td> <td>1110b</td> <td>20ns.</td> </tr> <tr> <td>0011b</td> <td>100ns.</td> <td>0111b</td> <td>60ns.</td> <td>1011b</td> <td>35ns.</td> <td>1111b</td> <td>15ns.</td> </tr> </tbody> </table> <p>If PowerStepDown or PowerStepUp are programmed to greater than 50ns, then the value applied to NB steps is clipped to 50ns. BIOS should program the PowerStepDown and PowerStepUp fields as follows:</p> <ul style="list-style-type: none"> <li>• Single-link processors: 1000b.</li> <li>• Multi-link processors where <a href="#">F0x68[DispRefModeEn]=1</a> or <a href="#">F0x[E4, C4, A4, 84][IsocEn]=1</a>: 1000b.</li> <li>• Other multi-link processors: PowerStepDown (ns) = PowerStepUp (ns) = 400 / # of cores in the node (e.g. 4 cores = 400 / 4 = 100ns = 0011b).</li> </ul>	<u>Bits</u>	<u>Time</u>	<u>Bits</u>	<u>Time</u>	<u>Bits</u>	<u>Time</u>	<u>Bits</u>	<u>Time</u>	0000b	400ns.	0100b	90ns.	1000b	50ns.	1100b	30ns.	0001b	300ns.	0101b	80ns.	1001b	45ns.	1101b	25ns.	0010b	200ns.	0110b	70ns.	1010b	40ns.	1110b	20ns.	0011b	100ns.	0111b	60ns.	1011b	35ns.	1111b	15ns.
<u>Bits</u>	<u>Time</u>	<u>Bits</u>	<u>Time</u>	<u>Bits</u>	<u>Time</u>	<u>Bits</u>	<u>Time</u>																																		
0000b	400ns.	0100b	90ns.	1000b	50ns.	1100b	30ns.																																		
0001b	300ns.	0101b	80ns.	1001b	45ns.	1101b	25ns.																																		
0010b	200ns.	0110b	70ns.	1010b	40ns.	1110b	20ns.																																		
0011b	100ns.	0111b	60ns.	1011b	35ns.	1111b	15ns.																																		
23:20	<p><b>PowerStepDown.</b> Read-write. Cold reset: 0000b. This specifies the rate at which blocks of core 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. See PowerStepUp for details.</p>																																								
19:18	Reserved.																																								

17:16	<p><b>LnkPllLock.</b> Read-write. Cold reset: 00b. This specifies the link PLL lock time applied when the link frequency is programmed to change during a link disconnect-reconnect sequence. The reconnect sequence is delayed to ensure that the PLL is locked. BIOS should set this field to 01b.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>PLL lock time</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>1 microsecond.</td> </tr> <tr> <td>01b</td> <td>10 microseconds.</td> </tr> <tr> <td>10b</td> <td>100 microseconds.</td> </tr> <tr> <td>11b</td> <td>1000 microseconds.</td> </tr> </tbody> </table>	Bits	PLL lock time	00b	1 microsecond.	01b	10 microseconds.	10b	100 microseconds.	11b	1000 microseconds.
Bits	PLL lock time										
00b	1 microsecond.										
01b	10 microseconds.										
10b	100 microseconds.										
11b	1000 microseconds.										
15	<p><b>StutterScrubEn: stutter mode scrub enable.</b> Revisions D0 and earlier: Reserved. Revision D1 and later: Read-write. Cold reset: 0. 1=Enable DRAM scrubbing when LDTSTOP_L is deasserted during stutter mode. One scrub request is sent for each LDTSTOP_L deassertion. See <a href="#">2.4.3.3.5 [BIOS Requirements to Initialize Message Triggered C1E]</a>.</p>										
14	<p><b>CacheFlushImmOnAllHalt: cache flush immediate on all halt.</b> Revisions D0 and earlier: Reserved. Revision D1 and later: Read-write. Cold reset: 0. 1=Flush the caches immediately when all cores in a package have halted. Setting this bit has no effect if <math>F3xDC[CacheFlushOnHaltCtl] = 0</math>. See <a href="#">2.4.3.3.5 [BIOS Requirements to Initialize Message Triggered C1E]</a>.</p>										
13	<p><b>MTC1eEn: message triggered C1E enable.</b> Revisions D0 and earlier: Reserved. Revision D1 and later: Read-write. Cold reset: 0. 1=Enables message triggered C1E. See <a href="#">2.4.3.3.5 [BIOS Requirements to Initialize Message Triggered C1E]</a>.</p>										
12	<p><b>ClkRampHystCtl: clock ramp hysteresis control.</b> Revision B: Reserved. Revision C: Read-write. Cold reset: 0b. When <math>F3x[84:80][CpuPrbEn]=0</math>, this field selects the time base for <math>F3xD4[ClkRampHystSel]</math>. 0=The time base is 320ns. 1=The time base is 1.28us. BIOS should leave this field in the reset state.</p>										
11:8	<p><b>ClkRampHystSel: clock ramp hysteresis select.</b> Read-write. Cold reset: 0000b. 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 <math>F3x[84:80][CpuPrbEn]=0</math> for the low-power state, 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. For revision B, the hysteresis time is encoded as <math>320ns * (1 + ClkRampHystSel)</math>. For revision C and later, the hysteresis time is encoded as <math>(\text{the time base specified by } F3xD4[ClkRampHystCtl]) * (1 + ClkRampHystSel)</math>.</li> <li>• If <math>F3x[84:80][CpuPrbEn]=1</math> for the low-power state, and for requests to change core P-states, 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=40ns to Fh=640ns, encoded as <math>40ns * (1 + ClkRampHystSel)</math>.</li> <li>• BIOS should set this field to 1111b.</li> </ul>										
7:6	Reserved.										

5	<b>NbFidEn: Northbridge frequency ID enable.</b> Read-write. Cold reset: value is the same as <a href="#">F3xA0[CofVidProg]</a> . This specifies the NB FID after warm or cold resets. 0=After a cold reset, the NB FID is 800 MHz, regardless of the state of NbFid. After a warm reset, the NB FID is the NB FID before the warm reset, regardless of the state of NbFid. 1=The NB FID is specified by NbFid. See <a href="#">2.4.2 [P-states]</a> .
4:0	<b>NbFid: Northbridge frequency ID.</b> Read-write. Cold reset: value varies by product. After a cold reset, this specifies the FID at which the NB is designed to operate. After a warm or cold reset, the NB FID may or may not be reflected in this field, based on the state of NbFidEn. The NB FID may be updated to the value of this field through a warm or cold reset if NbFidEn=1. If that has occurred, then the NB COF is specified by: <ul style="list-style-type: none"> <li>• <math>NB\ COF = 200\ MHz * (F3xD4[NbFid] + 4h) / (2^{MSRC001\_00[68:64][NbDid]})</math>.</li> </ul> This field must be programmed to the requirements specified in <a href="#">MSRC001\_0071[MaxNbFid]</a> and must be less than or equal to 1Bh, otherwise undefined behavior results. This field must be programmed to the same value for all nodes in the coherent fabric as specified by <a href="#">2.4.2.9 [BIOS Northbridge COF and VID Configuration]</a> . See <a href="#">2.4.2 [P-states]</a> . BIOS must not change the NbFid after enabling the DRAM controller.

### **F3xD8 Clock Power/Timing Control 1 Register**

The VID(s) are provided by the processor to the external voltage regulator(s). They can be altered through P-state changes.

Bits	Description
31:28	Reserved.
27:24	<b>ReConDel: link reconnect delay.</b> Read-write. Cold reset: 0. Specifies the approximate delay, in microseconds, from the deassertion of LDTSTOP_L until the link initialization process is allowed to start in Gen1 mode if <a href="#">F0x[E4, C4, A4, 84][LdtStopTriEn]=1</a> and <a href="#">F0x[18C:170][LS2En]=1</a> . The assertion of CTL is delayed until the specified time has elapsed. See <a href="#">2.7.6 [Link LDTSTOP_L Disconnect-Reconnect]</a> for information on when this is applied. The receiver is always enabled 1us after deassertion of LDTSTOP_L, regardless of the setting of this field or other delays in assertion of CTL. BIOS should program this field to 3h. 0h = 1.6us. 1h = 2us. ... 9h = 10us. All other values are reserved.
23	Reserved.
22:16	<b>AltVidStart: alternate VID start limit.</b> Read-write. Cold reset: 00h. Specifies the current VID limit required before driving out the alternate VID. The alternate VID (specified by <a href="#">F3xDC[AltVid]</a> ) is provided to the voltage regulator only if the voltage specified by the current VID is less than or equal to the voltage level specified by AltVidStart. See <a href="#">2.4.1.5 [Alternative Voltage (Altvid)]</a> . BIOS should program this to the same as <a href="#">MSRC001\_00[68:64][CpuVid]</a> of the P-state specified by <a href="#">F3xDC[PstateMaxVal]</a> . This should be done after all changes to <a href="#">MSRC001\_00[68:64][CpuVid]</a> and <a href="#">F3xDC[PstateMaxVal]</a> are complete.
14:8	<b>TdpVid: thermal design power VID.</b> Read-write. Cold reset: value varies by product. Specifies the voltage used to calculate the TDP of the processor. TdpVid is always encoded using <a href="#">Table 8</a> . See <a href="#">2.4.2.10 [Processor-Systemboard Power Removal Check]</a> .
7	Reserved.



6:4	<p><b>VSRampTime: voltage stabilization ramp time.</b> Read-write. Cold reset: 000b. Specifies the time to wait for voltage stabilization after each internal 7 bit VID increment (regardless of whether the SVI or PVI is used), if the voltage level is ramped. Refer to section 2.4.1.6 [VID Encodings] for internal 7 bit VID code to PVI VID encodings. If in SVI mode, this time measures the period from the end of each SVI command to the start of the next SVI command. See 2.4.1.8 [Hardware-Initiated Voltage Transitions].</p> <table border="1" data-bbox="280 405 1096 573"> <thead> <tr> <th><u>Bits</u></th> <th><u>Time</u></th> <th><u>Bits</u></th> <th><u>Time</u></th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>10 microseconds</td> <td>100b</td> <td>60 microseconds</td> </tr> <tr> <td>001b</td> <td>20 microseconds</td> <td>101b</td> <td>100 microseconds</td> </tr> <tr> <td>010b</td> <td>30 microseconds</td> <td>110b</td> <td>200 microseconds</td> </tr> <tr> <td>011b</td> <td>40 microseconds</td> <td>111b</td> <td>500 microseconds</td> </tr> </tbody> </table> <p>BIOS should set this field to 001b.</p>	<u>Bits</u>	<u>Time</u>	<u>Bits</u>	<u>Time</u>	000b	10 microseconds	100b	60 microseconds	001b	20 microseconds	101b	100 microseconds	010b	30 microseconds	110b	200 microseconds	011b	40 microseconds	111b	500 microseconds
<u>Bits</u>	<u>Time</u>	<u>Bits</u>	<u>Time</u>																		
000b	10 microseconds	100b	60 microseconds																		
001b	20 microseconds	101b	100 microseconds																		
010b	30 microseconds	110b	200 microseconds																		
011b	40 microseconds	111b	500 microseconds																		
3	Reserved.																				
2:0	<p><b>VSSlamTime: voltage stabilization slam time.</b> Read-write. Cold reset: 000b. Specifies the time to wait for voltage stabilization if a new VID is provided to the voltage regulator without ramping. See 2.4.1.8 [Hardware-Initiated Voltage Transitions]. If in SVI mode, then this time measures the period after the end of the SVI command.</p> <table border="1" data-bbox="280 804 1096 972"> <thead> <tr> <th><u>Bits</u></th> <th><u>Time</u></th> <th><u>Bits</u></th> <th><u>Time</u></th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>10 microseconds</td> <td>100b</td> <td>60 microseconds</td> </tr> <tr> <td>001b</td> <td>20 microseconds</td> <td>101b</td> <td>100 microseconds</td> </tr> <tr> <td>010b</td> <td>30 microseconds</td> <td>110b</td> <td>200 microseconds</td> </tr> <tr> <td>011b</td> <td>40 microseconds</td> <td>111b</td> <td>500 microseconds</td> </tr> </tbody> </table> <p>BIOS should program this field according to the following equation: <math>VSSlamTime = x * [P0\ voltage - Pmin\ voltage]</math> where <math>x = .4\mu s/mV</math> for server and desktop processors and <math>x = .2\mu s/mV</math> for mobile processors. The VSSlamTime value should be rounded up to the nearest programmable time if necessary. Lower VSSlamTime values are allowed if supported by the voltage regulator. Higher values may also be necessary based on voltage regulator requirements. See F3xDC[AltvidVSSlamTime, Slam-TimeMode].</p>	<u>Bits</u>	<u>Time</u>	<u>Bits</u>	<u>Time</u>	000b	10 microseconds	100b	60 microseconds	001b	20 microseconds	101b	100 microseconds	010b	30 microseconds	110b	200 microseconds	011b	40 microseconds	111b	500 microseconds
<u>Bits</u>	<u>Time</u>	<u>Bits</u>	<u>Time</u>																		
000b	10 microseconds	100b	60 microseconds																		
001b	20 microseconds	101b	100 microseconds																		
010b	30 microseconds	110b	200 microseconds																		
011b	40 microseconds	111b	500 microseconds																		



**F3xDC Clock Power/Timing Control 2 Register**

Reset: see field definitions.

Bits	Description																				
31:29	<p><b>AltvidVSSlamTime: altvid voltage stabilization slam time.</b> Revision C2 and earlier and revision D and later: Reserved. Revision C3: Read-write. Cold reset: 000b. Specifies the time to wait for voltage stabilization during altvid transitions if a new VID is provided to the voltage regulator without ramping. See SlamTimeMode and 2.4.3.5 [C5].</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Time</th> <th>Bits</th> <th>Time</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>&lt; 1 microsecond</td> <td>100b</td> <td>50 microseconds</td> </tr> <tr> <td>001b</td> <td>10 microseconds</td> <td>101b</td> <td>70 microseconds</td> </tr> <tr> <td>010b</td> <td>20 microseconds</td> <td>110b</td> <td>80 microseconds</td> </tr> <tr> <td>011b</td> <td>40 microseconds</td> <td>111b</td> <td>90 microseconds</td> </tr> </tbody> </table> <p>BIOS should program this field according to the following equation: <math>\text{AltvidVSSlamTime} = x * [\text{Pmin voltage} - \text{altvid voltage}]</math> where <math>x = .4\mu\text{s/mV}</math> for server and desktop processors and <math>x = .2\mu\text{s/mV}</math> for mobile processors. The AltvidVSSlamTime value should be rounded up to the nearest programmable time.</p>	Bits	Time	Bits	Time	000b	< 1 microsecond	100b	50 microseconds	001b	10 microseconds	101b	70 microseconds	010b	20 microseconds	110b	80 microseconds	011b	40 microseconds	111b	90 microseconds
Bits	Time	Bits	Time																		
000b	< 1 microsecond	100b	50 microseconds																		
001b	10 microseconds	101b	70 microseconds																		
010b	20 microseconds	110b	80 microseconds																		
011b	40 microseconds	111b	90 microseconds																		
28:27	<p><b>SlamTimeMode.</b> Revision C2 and earlier and revision D and later: Reserved. Revision C3: Read-write. Cold reset: 00b. Specifies the voltage stabilization slam time to use for transitions to lower performance P-states and altvid. Transitions to higher performance P-states always use F3xD8[VSSlamTime].</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>P-state transition down</th> <th>Altvid entry</th> <th>Altvid exit</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>F3xD8[VSSlamTime]</td> <td>&lt; 1 us</td> <td>AltvidVSSlamTime</td> </tr> <tr> <td>01b</td> <td>F3xD8[VSSlamTime]</td> <td>AltvidVSSlamTime</td> <td>AltvidVSSlamTime</td> </tr> <tr> <td>10b</td> <td>0 us</td> <td>&lt; 1 us</td> <td>AltvidVSSlamTime</td> </tr> <tr> <td>11b</td> <td>F3xD8[VSSlamTime]</td> <td>F3xD8[VSSlamTime]</td> <td>F3xD8[VSSlamTime]</td> </tr> </tbody> </table> <p>If CPUID Fn8000_0001_EBX[PkgType]=0010b or 0100b, BIOS should program this to 10b; else BIOS should leave this field at 00b. See 2.4.3.5 [C5]. If this field is programmed to a value less than 11b, the following condition must be true:</p> <ul style="list-style-type: none"> <li>F3xD8[AltVidStart]=MSRC001_00[68:64][CpuVid] of the P-state specified by F3xDC[PstateMaxVal].</li> </ul>	Bits	P-state transition down	Altvid entry	Altvid exit	00b	F3xD8[VSSlamTime]	< 1 us	AltvidVSSlamTime	01b	F3xD8[VSSlamTime]	AltvidVSSlamTime	AltvidVSSlamTime	10b	0 us	< 1 us	AltvidVSSlamTime	11b	F3xD8[VSSlamTime]	F3xD8[VSSlamTime]	F3xD8[VSSlamTime]
Bits	P-state transition down	Altvid entry	Altvid exit																		
00b	F3xD8[VSSlamTime]	< 1 us	AltvidVSSlamTime																		
01b	F3xD8[VSSlamTime]	AltvidVSSlamTime	AltvidVSSlamTime																		
10b	0 us	< 1 us	AltvidVSSlamTime																		
11b	F3xD8[VSSlamTime]	F3xD8[VSSlamTime]	F3xD8[VSSlamTime]																		
26	Reserved.																				
25:19	<p><b>CacheFlushOnHaltTmr: cache flush on halt timer.</b> Revision B: Reserved. Revision C and later: Read-write. Cold reset: 00h. When cache flushing during halt is enabled (see F3xDC[CacheFlushOnHaltCtl]) this field specifies how long each core needs to stay in halt before it flushes its caches.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Time</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td><math>\leq 5.12\mu\text{s}</math></td> </tr> <tr> <td>01h - 7Fh</td> <td><math>(\text{CacheFlushOnHaltTmr} * 10.24\mu\text{s}) - 5.12\mu\text{s} \leq \text{Time} \leq \text{CacheFlushOnHaltTmr} * 10.24\mu\text{s}</math></td> </tr> </tbody> </table> <p>If F3x[84:80][CpuPrbEn]=0 for the C1 state, only the time when the core is halted and has its clocks ramped up to service probes is counted. If F3x[84:80][CpuPrbEn]=1 for the C1 state, all of the time the core is halted is counted.</p> <p>BIOS should program this as follows:</p> <ul style="list-style-type: none"> <li>Revision DA-C and C3 single-link processors where CPUID Fn8000_0006_EDX[L3Size]=0: 4h.</li> <li>Revision E if CPB is enabled: Ch.</li> <li>All other revisions: 28h.</li> </ul>	Bits	Time	0h	$\leq 5.12\mu\text{s}$	01h - 7Fh	$(\text{CacheFlushOnHaltTmr} * 10.24\mu\text{s}) - 5.12\mu\text{s} \leq \text{Time} \leq \text{CacheFlushOnHaltTmr} * 10.24\mu\text{s}$														
Bits	Time																				
0h	$\leq 5.12\mu\text{s}$																				
01h - 7Fh	$(\text{CacheFlushOnHaltTmr} * 10.24\mu\text{s}) - 5.12\mu\text{s} \leq \text{Time} \leq \text{CacheFlushOnHaltTmr} * 10.24\mu\text{s}$																				

18:16	<p><b>CacheFlushOnHaltCtl: cache flush on halt control.</b> Revision B: Reserved. Revision C and later: Read-write. Cold reset: 000b. Specifies whether caches are flushed during halt and what core clock divisor is used after the caches have been flushed. BIOS should leave this field at 000b for revision C2 single-core processors. For all other processors BIOS should program this to 111b.</p> <table> <thead> <tr> <th>Bits</th> <th>Divisor</th> <th>Bits</th> <th>Divisor</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Cache flush on halt disabled.</td> <td>100b</td> <td>Divide-by 16.</td> </tr> <tr> <td>001b</td> <td>Divide-by 2.</td> <td>101b</td> <td>Divide-by 128.</td> </tr> <tr> <td>010b</td> <td>Divide-by 4.</td> <td>110b</td> <td>Divide-by 512.</td> </tr> <tr> <td>011b</td> <td>Divide-by 8.</td> <td>111b</td> <td>Turn off clocks.</td> </tr> </tbody> </table> <p>See [The ACPI Power State Control Registers] F3x[84:80] for clock divisor specifications that are in effect during halt before the caches have been flushed.</p>	Bits	Divisor	Bits	Divisor	000b	Cache flush on halt disabled.	100b	Divide-by 16.	001b	Divide-by 2.	101b	Divide-by 128.	010b	Divide-by 4.	110b	Divide-by 512.	011b	Divide-by 8.	111b	Turn off clocks.
Bits	Divisor	Bits	Divisor																		
000b	Cache flush on halt disabled.	100b	Divide-by 16.																		
001b	Divide-by 2.	101b	Divide-by 128.																		
010b	Divide-by 4.	110b	Divide-by 512.																		
011b	Divide-by 8.	111b	Turn off clocks.																		
15	Reserved.																				
14:12	<p><b>NbsynPtrAdj: NB/core synchronization FIFO pointer adjust.</b> Read-write. Cold reset: 0h. 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. This field 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 this field represents one clock cycle of whichever is the slower clock (longer period) between the NB clock and the core clock. After writing to this field, the new values are applied after a warm reset. If (Revision B and earlier)    (BL-C)    (DA-C)    (Revision RB-C3 &amp;&amp; F3x1F0[NbPstate]! = 000b), BIOS should program this field to 5h; otherwise BIOS should program this field to 6h. Values less than the recommended value are allowed; values greater than the recommended value are illegal.</p> <table> <tbody> <tr> <td>0h</td> <td>Position the read pointer 0 clock cycles closer to the write pointer.</td> </tr> <tr> <td>1h</td> <td>Position the read pointer 1 clock cycles closer to the write pointer.</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>6h</td> <td>Position the read pointer 6 clock cycles closer to the write pointer.</td> </tr> </tbody> </table>	0h	Position the read pointer 0 clock cycles closer to the write pointer.	1h	Position the read pointer 1 clock cycles closer to the write pointer.	...	...	6h	Position the read pointer 6 clock cycles closer to the write pointer.												
0h	Position the read pointer 0 clock cycles closer to the write pointer.																				
1h	Position the read pointer 1 clock cycles closer to the write pointer.																				
...	...																				
6h	Position the read pointer 6 clock cycles closer to the write pointer.																				
11	Reserved.																				
10:8	<p><b>PstateMaxVal: P-state maximum value.</b> Read-write. Cold reset: specified by the reset state of MSRC001_00[68:64][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 PstateMaxVal=1; if PstateEn is not set in any of these MSRs, then PstateMaxVal=0). This specifies the highest P-state value (lowest performance state) supported by the hardware. See MSRC001_0061[PstateMaxVal]. This field uses hardware P-state numbering. See 2.4.2.1.2.2 [Hardware P-state Numbering].</p>																				
7	Reserved.																				
6:0	<p><b>AltVid: alternate VID.</b> Read-write. Cold reset: value varies by product. If F3xA0[PviMode]=0, this specifies the VID driven to the VDD power plane while in the low-power state, as specified by [The ACPI Power State Control Registers] F3x[84:80][AltVidEn]. This field is required to be programmed as specified by MSRC001_0071[MaxVid and MinVid]. See 2.4.1.5 [Alternative Voltage (Altvid)].</p>																				

### F3xE4 Thermtrip Status Register

Reset: 0000 0000h, except bits[14:8, 5, 3 and 1]; see below.

Bits	Description
31	<b>SwThermtp: software THERMTRIP.</b> Write-1-only. 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.
30:15	Reserved.

14:8	<b>DiodeOffset.</b> Read-only. Reset: value varies by product. This field is used to specify the correction value applied to thermal diode measurements. See <a href="#">2.10.2 [Thermal Diode]</a> . It is encoded as follows: 00h is Diode not supported. 01h to 3Fh: correction = +11C - DiodeOffset, or {01h to 3Fh} = {+10C to -52C}. 40h to 7Fh: undefined.
7:6	Reserved.
5	<b>ThermtpEn: THERMTRIP enable.</b> Read-only. 1=The THERMTRIP state as specified in section <a href="#">2.10.3.3 [THERMTRIP]</a> is supported by the processor.
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).
2	Reserved.
1	<b>Thermtp: THERMTRIP.</b> Read-only. Cold reset: 0. 1=The processor has entered the THERMTRIP state.
0	Reserved.

### F3xE8 Northbridge Capabilities Register

All fields are read-only. Unless otherwise specified, 1=The feature is supported by the processor; 0=The feature is not supported.

Bits	Description
31:30	<b>IntNodeNum.</b> Revision C and prior revisions: Reserved. Revision D and later: Specifies the internal node number for multi-node processors. 00b node 0. 01b node 1. 1xb reserved
29	<b>MultiNodeCpu.</b> Revision C and prior revisions: Reserved. Revision D and later: 1=The processor is a multi-node processor. 0=The processor is a single-node processor.
25	<b>L3Capable.</b> 1=Specifies that an L3 cache is present. See <a href="#">CPUID Fn8000_0006_EDX</a> .
24	Reserved.
23:20	<b>UnGangEn: link unganging enabled.</b> 0=Link is forced into the ganged state and may not be placed into the unganging state. 1=Unganging is supported. Bit[20] applies to link 0; bit[21] applies to link 1; bit[22] applies to link 2; bit[23] applies to link 3. See <a href="#">2.7 [Links]</a> .
19	Reserved.
18:16	<b>MpCap: MP capability.</b> Specifies the maximum number of processors supported as follows: 111b 1 processor. 110b 2 processors. 101b 4 processors. 000b 8 processors. All other values are reserved.
15	<b>CmpCap[2]: CMP capable[2].</b> Revision C and earlier: Reserved. Revision D and later: See CmpCap[1:0].
14	<b>Multiple VID plane capable.</b>
13:12	<b>CmpCap[1:0]: CMP capable[1:0].</b> This field along with CmpCap[2] specifies the number of cores present on the node. 000b=1; 001b=2; 010b=3; 011b=4; 100b=5; 101b=6. The value of CmpCap is unaffected by <a href="#">F3x190[DisCore]</a> .
11	<b>LnkRtryCap. Link error-retry capable.</b>

10	<b>HTC capable.</b> This affects F3x64 and F3x68.																				
9	<b>SVM capable.</b>																				
8	<b>MctCap: memory controller (on the processor) capable.</b>																				
7:5	<b>DdrMaxRate.</b> Specifies the maximum DRAM data rate that the processor is designed to support. <table border="1"> <thead> <tr> <th>Bits</th> <th>DDR limit</th> <th>Bits</th> <th>DDR limit</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>No limit</td> <td>100b</td> <td>800 MT/s</td> </tr> <tr> <td>001b</td> <td>Reserved</td> <td>101b</td> <td>667 MT/s</td> </tr> <tr> <td>010b</td> <td>1333 MT/s</td> <td>110b</td> <td>533 MT/s</td> </tr> <tr> <td>011b</td> <td>1067 MT/s</td> <td>111b</td> <td>400 MT/s</td> </tr> </tbody> </table>	Bits	DDR limit	Bits	DDR limit	000b	No limit	100b	800 MT/s	001b	Reserved	101b	667 MT/s	010b	1333 MT/s	110b	533 MT/s	011b	1067 MT/s	111b	400 MT/s
Bits	DDR limit	Bits	DDR limit																		
000b	No limit	100b	800 MT/s																		
001b	Reserved	101b	667 MT/s																		
010b	1333 MT/s	110b	533 MT/s																		
011b	1067 MT/s	111b	400 MT/s																		
4	<b>Chipkill ECC capable.</b>																				
3	<b>ECC capable.</b>																				
2	<b>Eight-node multi-processor capable.</b>																				
1	<b>Dual-node multi-processor capable.</b>																				
0	<b>DctDualCap: two-channel DRAM capable (i.e., 128 bit).</b> 0=Single channel (64-bit) only.																				

### F3xF0 DEV Capability Header Register

See 2.6.3 [DMA Exclusion Vectors (DEV)]. Note: if SVM is not supported, as specified by F3xE8[SVM Capable], then this register is reserved. See Table 2 for revision specific DEV support.

DMA Exclusion Vectors (DEV) are contiguous arrays of bits in physical memory. Each bit in the DEV table represents a 4KB page of physical memory; the DEV applies to accesses that target system memory and MMIO, but not to accesses within the GART aperture (see 2.6.2 [The GART]). The DEV table is packed as follows: bit[0] of byte 0 (pointed to by the DEV table base address, F3xF8\_x0 and F3xF8\_x1) controls the first 4K bytes of physical memory (starting at address 00\_0000\_0000h); bit[1] of byte 0 controls the second 4K bytes of physical memory; etc. When a DEV table bit is set to one, accesses to that physical page by external DMA devices is not allowed. If an external device attempts to access a protected physical page, then the processor master aborts the request.

In addition, the processor supports multiple protection domains. There is a DEV table for each protection domain. Link-defined UnitIDs may be assigned to the DEV of a specific protection domain through F3xF8\_x2. DEV table walks for each protection domain are cached in the NB to reduce the number DEV table access to system memory.

The DEV function is configured through F3xF0, F3xF4, F3xF8, and an array of registers called F3xF8\_x[7:0], which are defined following F3xF8. [The DEV Function/Index Register] F3xF4 and [The DEV Data Port] F3xF8 are used to access F3xF8\_x[7:0]. The register number (i.e., the number that follows “x” in the register mnemonic) is specified by F3xF4[DevFunction]. In addition, F3xF8\_x0, F3xF8\_x1, and F3xF8\_x2 are each instantiated multiplied times, indexed by F3xF4[DevIndex]. To access the registers, F3xF4[DevFunction and DevIndex] are programmed to point to the appropriate register and the read or write access is directed at F3xF8.

Bits	Description
31:22	Reserved.
21	<b>IntCap: interrupt reporting capability.</b> Read-only. Reset: 0. 0=Indicates that interrupt reporting of DEV protection violations is not present on this device.

20	<b>MceCap: MCE reporting capability.</b> Read-only. Reset: 1. Indicates that machine check architecture reporting of DEV protection violations is present on this device.
19	Reserved.
18:16	<b>CapType: DEV capability block type.</b> Read-only. Reset: 000b. Specifies the layout of the Capability Block.
15:8	<b>CapPtr: capability pointer.</b> Read-only. Reset: 00h. Indicates that this is the last capability block.
7:0	<b>CapId: capability ID.</b> Read-only. Reset: 0Fh. Indicates a DEV capability block.

### F3xF4 DEV Function/Index Register

Reset: 0000 0000h. Note: if SVM is not supported, as specified by F3xE8[SVM Capable], then this register is reserved.

Bits	Description
31:16	Reserved.
15:8	<b>DevFunction.</b> Read-write. See F3xF0 for details. Valid values for this field are 00h through 07h.
7:0	<b>DevIndex.</b> Read-write. See F3xF0 for details. Valid values for this field are (1) 00h through (F3xF8_x3[NDomains] - 1) when either F3xF8_x0 or F3xF8_x1 are being accessed and (2) 00h through (F3xF8_x3[NMaps] - 1) when F3xF8_x2 is being accessed; this field is ignored for accesses to all other DEV configuration registers.

### F3xF8 DEV Data Port

Note: if SVM is not supported, as specified by F3xE8[SVM Capable], then this location and registers F3xF8\_x[7:0] are reserved. See F3xF0 for details about this port.

### F3xF8\_x0 DEV Base Address/Limit Low Register

Reset: 0000 0000h. This register is instantiated multiple times, specified by F3xF8\_x3[NDomains]. Each instantiation corresponds to a protection domain number, identical to F3xF4[DevIndex], which is the index to the instantiation. See F3xF0 for more details.

Bits	Description
31:12	<b>BaseAddress[31:12]: DEV table base address bits[31:12].</b> Read-write. These bits are combined with F3xF8_x1[BaseAddress[47:32]] to specify the base address of the DEV table. The DEV table is required to be in either non-cacheable or write-through memory. Placing DEV tables in MMIO space is not supported. If any part of the DEV table is in other than system memory, then undefined behavior results.
11:7	Reserved.
6:2	<b>Size: DEV table size.</b> Read-write. These bits specify the size of the memory region that the DEV table covers, 4GB*(2^Size). The corresponding DEV table size is 128KB*(2^Size).
1	<b>Protect: protect out-of-range addresses.</b> Read-write. 0=DMA accesses to addresses that are outside the range covered by the DEV table are allowed. 1=DMA accesses to addresses that are outside the range covered by the DEV table are protected.
0	<b>Valid: DEV table valid.</b> Read-write. 1=The DEV table for the protection domain specified by F3xF4[DevIndex] is enabled. 0=The DEV table is not enabled; all IO accesses from devices assigned to the corresponding protection domain are allowed.

### F3xF8\_x1 DEV Base Address/Limit High Register

Reset: 0000 0000h. This register is instantiated multiple times, specified by [F3xF8\\_x3\[NDomains\]](#). Each instantiation corresponds to a protection domain number, identical to [F3xF4\[DevIndex\]](#), which is the index to the instantiation. See [F3xF0](#) for more details.

Bits	Description
31:16	Reserved.
15:0	<b>BaseAddress[47:32]: DEV table base address bits[47:32].</b> Read-write. See <a href="#">F3xF8_x0[BaseAddress]</a> .

### F3xF8\_x2 DEV Map Register

Reset: 0000 0000h. This register is instantiated multiple times, specified by [F3xF8\\_x3\[NMaps\]](#). See [F3xF0](#) for more details. Referencing the fields of this register, if [Valid\[1:0\]](#) is set, then the address of DMA requests received by the processor from an IO link of bus number [BusNu](#) and with a [UnitID](#) of [Unit\[1:0\]](#) are checked against the DEV table of protection domain number [Dom\[1:0\]](#) to determine if the transaction is allowed. If the [UnitID](#) and [BusNu](#) of the request do not match any of these registers, then the address of the request is checked against the DEV table of protection domain 0 to determine if the transaction is allowed. A [UnitID](#) can only be assigned to one protection domain. If a [UnitID](#) is assigned to more than one protection domain the results are undefined. Note: if [UnitID](#) clumping is employed through [F0x\[11C, 118, 114, 110\]](#) and [F0x\[12C, 128, 124, 120\]](#), then only the base [UnitID](#) of each clump should be programmed into the [Unit0/Unit1](#) fields of this register; otherwise undefined behavior results.

Bits	Description
31:26	<b>Dom1: protection domain 1.</b> 3 LSBs are read-write; 3 MSBs are read-only, 000b. This is the protection domain number assigned to <a href="#">Unit1</a> .
25:20	<b>Dom0: protection domain 0.</b> 3 LSBs are read-write; 3 MSBs are read-only, 000b. This is the protection domain number assigned to <a href="#">Unit0</a> .
19:12	<b>BusNu: bus number.</b> Read-write.
11	<b>Valid1: UnitID 1 valid.</b> Read-write. 1=Enable DEV checking for <a href="#">Unit1</a> and <a href="#">Dom1</a> .
10:6	<b>Unit1: IO link UnitID 1.</b> Read-write.
5	<b>Valid0: UnitID 0 valid.</b> Read-write. 1=Enable DEV checking for <a href="#">Unit0</a> and <a href="#">Dom0</a> .
4:0	<b>Unit0: IO link UnitID 0.</b> Read-write.

### F3xF8\_x3 DEV Capabilities Register

Bits	Description
31:24	Reserved.
23:16	<b>NMaps: number of map registers implemented.</b> Read-only, 04h. Specifies the number of instantiations of <a href="#">F3xF8_x2</a> .
15:8	<b>NDomains: number of protection domains implemented.</b> Read-only, 08h. Specifies the number of protection domains and the number of instantiations of <a href="#">F3xF8_x0</a> and <a href="#">F3xF8_x1</a> .
7:0	<b>Revision: DEV register-set revision number.</b> Read-only, 00h.



**F3xF8\_x4 DEV Control Register**

Reset: 0000 0002h.

Bits	Description
31:7	Reserved.
6	<b>DevTblWalkPrbDis: DEV table walk probe disable.</b> Read-write. 1=Disable probing of CPU caches during DEV table walks. This bit may be set to improve DEV cache table walk performance when the DEV is in non-cacheable or write-through memory.
5	<b>SIDev: secure loader DEV protection enable.</b> Read-write; set-by-hardware. This bit is set by hardware after an SKINIT instruction. 1=The memory region associated with the SKINIT instruction is protected from DMA access.
4	<b>DevInv: invalidate DEV cache.</b> Read; write-1-only. 1=Invalidate the DEV table-walk cache. This bit is cleared by hardware when invalidation is complete.
3	<b>MceEn: MCE reporting enable.</b> Read-write. 1=Enable logging and reporting of DEV protection violations through a machine check exception.
2	<b>IoDis: upstream IO disable.</b> Read-write; set-by-hardware. This bit is set by hardware after an SKINIT instruction. 1=Upstream IO-space accesses are regarded as DEV protection violations.
1	Reserved. Read-write.
0	<b>DevEn: DEV enable.</b> Read-write. 1=Enables DMA exclusion vector protection.

**F3xF8\_x5 DEV Error Status Register**

Cold reset: 0000 0000h. This register logs DEV protection violations. Bits[7:0], [ErrTypeDest, ErrTypeSrc, ErrTypeAccType], together form the error type field. When a DEV protection violation occurs, then ErrVal is set, the error type is logged, and, if there is an address associated with the transaction, ErrAddrVal is set and the address is recorded in [F3xF8\\_x6](#) and [F3xF8\\_x7](#).

Bits	Description								
31	<b>ErrVal: error valid.</b> Read-write; set-by-hardware. 1=A valid DEV protection violation has been logged in this register.								
30	<b>ErrOver: error overflow.</b> Read-write; set-by-hardware. 1=A DEV protection violation was detected while ErrVal was set for a prior violation. DEV protection violations detected while ErrVal is set are not logged in this register.								
29	<b>ErrAddrVal: error address valid.</b> Read-write; set-by-hardware. 1=The address saved in <a href="#">F3xF8_x6</a> and <a href="#">F3xF8_x7</a> is the address associated with the error.								
28:24	Reserved.								
23:16	<b>ModelSpecErr: model specific error.</b> Read-only, 00h.								
15:8	Reserved.								
7:5	<b>ErrCodeDest: error code destination.</b> Read-write; set-by-hardware. Specifies the destination of the transaction that resulted in the protection violation. <table style="width: 100%; border: none;"> <tr> <td style="width: 50%;">000b = Generic (or could not be determined)</td> <td style="width: 50%;">100b = IO space</td> </tr> <tr> <td>001b = DRAM</td> <td>101b = Configuration</td> </tr> <tr> <td>010b = MMIO</td> <td>110b = reserved</td> </tr> <tr> <td>011b = reserved</td> <td>111b = reserved</td> </tr> </table>	000b = Generic (or could not be determined)	100b = IO space	001b = DRAM	101b = Configuration	010b = MMIO	110b = reserved	011b = reserved	111b = reserved
000b = Generic (or could not be determined)	100b = IO space								
001b = DRAM	101b = Configuration								
010b = MMIO	110b = reserved								
011b = reserved	111b = reserved								



4:2	<b>ErrCodeSrc: error code source.</b> Read-write; set-by-hardware. Specifies the source of the transaction that resulted in the protection violation. 000b = Generic (or could not be determined)      010b = IO device 001b = CPU      011b - 111b = reserved
1:0	<b>ErrCodeAccType: error code access type.</b> Read-write; set-by-hardware. Specifies the access type of the transaction that resulted in the protection violation. 00b = Generic (or could not be determined)      10b = Write 01b = Read      11b = Read-modify-write

### F3xF8\_x6 DEV Error Address Low Register

Cold reset: 0000 0000h.

Bits	Description
31:2	<b>ErrAddr: error address bits[31:2].</b> Read-write; set-by-hardware. See <a href="#">F3xF8_x5</a> for details.
1:0	Reserved.

### F3xF8\_x7 DEV Error Address High Register

Cold reset: 0000 0000h.

Bits	Description
31:16	Reserved.
15:0	<b>ErrAddr: error address bits[47:32].</b> Read-write; set-by-hardware. See <a href="#">F3xF8_x5</a> for details.

### F3xFC CPUID Family/Model Register

These values are identical to the values read out through [CPUID Fn\[8000\\_0001, 0000\\_0001\]\\_EAX](#); see that register for details.

Bits	Description
31:28	Reserved.
27:20	<b>ExtFamily: extended family.</b> Read-only.
19:16	<b>ExtModel: extended model.</b> Read-only.
15:12	Reserved.
11:8	<b>BaseFamily.</b> Read-only.
7:4	<b>BaseModel.</b> Read-only.
3:0	<b>Stepping.</b> Read-only.

### F3x10C Boost Offset

Revision C and earlier: Reserved.

Revision D: All fields are read-only.

Bits	Description
31:0	Reserved.

Revision E: All fields are read-only.

Bits	Description
31:12	Reserved.
11:10	<b>AsymmetricBoostCore5.</b> See AsymmetricBoostCore0.
9:8	<b>AsymmetricBoostCore4.</b> See AsymmetricBoostCore0.
7:6	<b>AsymmetricBoostCore3.</b> See AsymmetricBoostCore0.
5:4	<b>AsymmetricBoostCore2.</b> See AsymmetricBoostCore0.
3:2	<b>AsymmetricBoostCore1.</b> See AsymmetricBoostCore0.
1:0	<b>AsymmetricBoostCore0.</b> Read-only. Reset: value varies by product. Specifies the FID offset for the core when in the boosted state. This value is added to MSRC001_0064[CpuFid]. See 2.4.2.8 [BIOS Configuration for Asymmetric Boost]. See MSRC001_00[68:64][CpuFid] for encoding.

### F3x138 DCT0 Bad Symbol Identification Register

Revision C and earlier: Reserved. Revision D and later: Reset: 0000 0000h. All bits are read-write.

This register controls bad symbol identification for DCT0. This register is set by software and read by hardware to supplement ECC checks during chipkill conditions. The MCT uses this information during fetches to determine if errors are correctable or uncorrectable. See 2.12.2.2 [Software Managed Bad Symbol Identification] for additional details.

Bits	Description
31	<b>BadDramAllCsEn: Bad DRAM on all chip selects.</b> Indicates that the symbol number specified in BadDramSymbol applies to all chip selects. This most likely indicates a fault outside the DRAM, since it affects all ranks.
30:12	Reserved.
11	<b>BadDramCsVal: Bad DRAM chip select valid.</b> Indicates that BadDramCs and BadDramSymbol contain valid information.
10:9	Reserved
8:4	<b>BadDramSymbol.</b> Indicates the bad symbol number within the rank. Values 0h through Fh correspond to data byte 0h through Fh, respectively. Value 10h corresponds to ECC byte 0, and value 11h corresponds to ECC byte 1.
3:0	<b>BadDramCs: Bad DRAM chip select.</b> Indicates the chip select value which is known bad. This chip select value identifies the rank in error.

### F3x13C DCT1 Bad Symbol Identification Register

Revision C and earlier: Reserved. Revision D and later: This register controls bad symbol identification for DCT1. See F3x138 for register definition.

### F3x140 SRI to XCS Token Count Register

F3x140, F3x144, and F3x1[54, 50, 4C, 48] specify the number of XCS (XBAR command scheduler) entries assigned to each virtual channel within each source port. See 2.6.1 [Northbridge (NB) Architecture]. The totals of SRI, MCT and the links must not exceed the number of XCS entries. The default totals are:

- SRI: 10.
- MCT: 6.
- Link: 10 \* 4 (one group per link).

- Total: 56, which is the total number of entries supported by XCS.

Note that the defaults for **F3x140**, **F3x1[54, 50, 4C, 48]**, and **F3x158** do not allocate any tokens in the isochronous channel. If isochronous flow control mode (IFCM) is enabled (**F0x[E4, C4, A4, 84][IsocEn]**) or display refresh mode is enabled (**F0x68[DispRefModeEn]**), then the XCS token counts must be changed. Notes on these modes:

- If IFCM is enabled on any link, then the **F3x140[IsocReqTok, IsocPreqTok, and IsocRspTok]** must each be non-zero. If display refresh mode is enabled, **F3x140[IsocReqTok and IsocRspTok]** must be non-zero, and **F3x140[IsocPreqTok]** must be non-zero if **F3x158[LnkToXcsDRToken]** is non-zero. This requires tokens to be reduced elsewhere to avoid exceeding the 56 token maximum. Note that links which are not connected or links which are ganged include excess tokens which may be used for this purpose.
- If IFCM is enabled on any link, then it may be advantageous to allocate isochronous tokens to that link/sub-link in **F3x1[54, 50, 4C, 48]**. However this would result in excessive tokens for a fully populated system, especially if the links are unganged. To account for this, the processor supports IFCM being enabled on a link without allocating dedicated isochronous XCS tokens. In this case:
  - The isochronous channel uses the base channel tokens.
  - The isochronous channel has priority access to these tokens.
- If an IOMMU is present in the system, **F3x1[54, 50, 4C, 48][IsocReqTok]** for all coherent links in the system must be non-zero and **F3x1[54, 50, 4C, 48][IsocReqTok]** for the IO link with the IOMMU must be non-zero.
- In display refresh mode, **F3x1[54, 50, 4C, 48][IsocReqTok]** and **F3x1[54, 50, 4C, 48][IsocPreqTok]** for the enabled link and **F3x158[LnkToXcsDRToken]** must be non-zero.

Updates to this register do not take effect until after a warm reset.

See 2.6.4.2.5.1 [Recommended Buffer Count Settings for Single Link Processors] and 2.6.4.2.5.2 [Recommended Buffer Count Settings for Multi-link Processors].

Bits	Description
31:24	Reserved.
23:20	<b>FreeTok: free tokens.</b> Read-write. Cold reset: 2. 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. Cold reset: 0.
15:14	<b>IsocPreqTok: isochronous posted request tokens.</b> Read-write. Cold reset: 0.
13:12	<b>IsocReqTok: isochronous request tokens.</b> Read-write. Cold reset: 0.
11:10	<b>DnRspTok: downstream response tokens.</b> Read-write. Cold reset: 1.
9:8	<b>UpRspTok: upstream response tokens.</b> Read-write. Cold reset: 2.
7:6	<b>DnPreqTok: downstream posted request tokens.</b> Read-write. Cold reset: 1.
5:4	<b>UpPreqTok: upstream posted request tokens.</b> Read-write. Cold reset: 1.
3:2	<b>DnReqTok: downstream request tokens.</b> Read-write. Cold reset: 1.
1:0	<b>UpReqTok: upstream request tokens.</b> Read-write. Cold reset: 2.

### **F3x144 MCT to XCS Token Count Register**

See **F3x140** for more information. Updates to **F3x144** do not take effect until after a warm reset.

Bits	Description
31:8	Reserved.
7:4	<b>ProbeTok: probe tokens.</b> Read-write. Cold reset: 3.
3:0	<b>RspTok: response tokens.</b> Read-write. Cold reset: 3.

### **F3x1[54, 50, 4C, 48] Link to XCS Token Count Registers**

F3x148 applies to link 0; F3x14C applies to link 1; F3x150 applies to link 2; F3x154 applies to link 3. See [F3x140](#) for more information. The cold reset default value for some of the fields of this register vary based on the ganged/unganged state specified by [F0x\[18C:170\]\[Ganged\]](#). Most of the fields in this register are duplicated for each sublink; if the link is ganged, then the sublink 0 fields apply and the sublink 1 fields should be 0. Updates to [F3x1\[54, 50, 4C, 48\]](#) do not take effect until after a warm reset. See [2.7.1.3.1 \[Unused Links\]](#).

Bits	Description
31:30	<b>FreeTok[3:2]: free tokens.</b> Read-write. See FreeTok[1:0] below.
29	Reserved.
28	<b>IsocRspTok1: isochronous response tokens sublink 1.</b> Read-write. Cold reset: 0.
27	Reserved.
26	<b>IsocPreqTok1: isochronous posted request tokens sublink 1.</b> Read-write. Cold reset: 0.
25	Reserved.
24	<b>IsocReqTok1: isochronous request tokens sublink 1.</b> Read-write. Cold reset: 0.
23:22	<b>ProbeTok1: probe tokens sublink 1.</b> Read-write. Cold reset: 0 ganged, 1 unganged.
21:20	<b>RspTok1: response tokens sublink 1.</b> Read-write. Cold reset: 0 ganged, 1 unganged.
19:18	<b>PReqTok1: posted request tokens sublink 1.</b> Read-write. Cold reset: 0 ganged, 1 unganged.
17:16	<b>ReqTok1: request tokens sublink 1.</b> Read-write. Cold reset: 0 ganged, 1 unganged.
15:14	<b>FreeTok[1:0]: free tokens.</b> Read-write. Cold reset: 0010b (for FreeTok[3:0]). FreeTok[3:0] is 4-bit field composed of {FreeTok[3:2], FreeTok[1:0]} in this register. If the link is unganged, the free tokens are shared between the two sublinks.
13:12	<b>IsocRspTok0: isochronous response tokens sublink 0.</b> Read-write. Cold reset: 0.
11:10	<b>IsocPreqTok0: isochronous posted request tokens sublink 0.</b> Read-write. Cold reset: 0.
9:8	<b>IsocReqTok0: isochronous request tokens sublink 0.</b> Read-write. Cold reset: 0.
7:6	<b>ProbeTok0: probe tokens sublink 0.</b> Read-write. Cold reset: 2 ganged, 1 unganged.
5:4	<b>RspTok0: response tokens sublink 0.</b> Read-write. Cold reset: 2 ganged, 1 unganged.
3:2	<b>PReqTok0: posted request tokens sublink 0.</b> Read-write. Cold reset: 2 ganged, 1 unganged.
1:0	<b>ReqTok0: request tokens sublink 0.</b> Read-write. Cold reset: 2 ganged, 1 unganged.

### **F3x158 Link to XCS Token Count Registers**

See [F3x140](#) for more information. Updates to [F3x158](#) do not take effect until after a warm reset.

Bits	Description
------	-------------

31:4	Reserved.
3:0	<b>LnkToXcsDRToken: display refresh tokens all links.</b> Read-write. Cold reset: 0.

### F3x1[78, 70, 68, 60] NB Machine Check Misc (Thresholding) Registers

These registers may also be accessed through [MSR0000\\_0413](#) and [MSRC000\\_04\[0A:08\]](#). These registers are associated with the following error types as specified by the Error Threshold Group in [Table 93](#) of [\[The MCA NB Status Low Register\] F3x48\[ErrorCode\]](#):

- F3x160 (MSR0000\_0413): DRAM.
- F3x168 (MSRC000\_0408): Link.
- F3x170 (MSRC000\_0409): L3 Cache. If the product does not include an L3 cache, per [\[The L2/L3 Cache and L2 TLB Identifiers\] CPUID Fn8000\\_0006\\_EDX](#), then the Valid and CntP bits are both 0 and the register logs no information.
- F3x178 (MSRC000\_040A): RAZ.

For general information on error thresholding, see [2.12.1.6 \[Error Thresholding\]](#).

Bits	Description
31	<b>Valid.</b> Read-only from configuration space; read-only or read-write from MSR space based on <a href="#">MSRC001_0015[McStatusWrEn]</a> . Reset: 1. 1=A valid CntP field is present in this register.
30	<b>CntP: counter present.</b> Read-only from configuration space; read-only or read-write from MSR space based on <a href="#">MSRC001_0015[McStatusWrEn]</a> . Reset: 1. 1=A valid threshold counter is present. This bit is affected by <a href="#">MSRC001_0015[McStatusWrEn]</a> .
29	<b>Locked.</b> Read-only from configuration space; read-only or read-write from MSR space based on <a href="#">MSRC001_0015[McStatusWrEn]</a> . Reset: 0. This bit is set by BIOS to indicate that this register is not available for OS use. When this bit is set, write to bits[28:0] of this register are ignored. BIOS should set this bit if IntType is set to SMI. Note: when <a href="#">MSRC001_0015[McStatusWrEn]</a> is set, MSR writes to this register update all bits, regardless of the state of the Locked bit.
28:24	Reserved.
23:20	<b>LvtOffset: LVT offset.</b> Read-write (see the <a href="#">F3x1[78, 70, 68, 60][Locked]</a> ). Reset: 0h. This field specifies the address of the LVT entry in the APIC registers as follows: LVT address = (LvtOffset shifted left 4 bits) + 500h (see <a href="#">APIC[530:500]</a> ). Only values 0 through 3 are valid; all others reserved.
19	<b>CntEn: counter enable.</b> Read-write (see the <a href="#">F3x1[78, 70, 68, 60][Locked]</a> ). Reset: 0. 1=Counting of errors specified by Error Threshold Group (see above) is enabled.
18:17	<b>IntType: interrupt type.</b> Read-write (see the <a href="#">F3x1[78, 70, 68, 60][Locked]</a> ). Cold reset: 0. This field specifies the type of interrupt signaled when Ovrflw is set. 00b = No Interrupt. 01b = APIC based interrupt (see LvtOffset above) to all cores. 10b = SMI trigger event (always routed to CpuCoreNum 0, as defined in section <a href="#">2.9.2 [CPU Cores and Downcoring]</a> ); see <a href="#">2.13.2.3 [SMI Sources And Delivery]</a> . 11b = Reserved.
16	<b>Ovrflw: overflow.</b> Read-write (see the <a href="#">F3x1[78, 70, 68, 60][Locked]</a> ); set-by-hardware. Cold reset: X. This bit is set by hardware when ErrCnt transitions from FFEh to FFFh. When this bit is set, the interrupt selected by the IntType field is generated.

15:12	Reserved.
11:0	<b>ErrCnt: error counter.</b> Read-write (see the F3x1[78, 70, 68, 60][Locked]). Cold reset: X. This is written by software to set the starting value of the error counter. This is incremented by hardware when errors are logged. When this counter overflows, it stays at FFFh (no rollover). To set the threshold value, software should subtract the desired error count (the number of errors necessary in order for an interrupt to be taken) from FFFh and write the result into this field.

### F3x180 Extended NB MCA Configuration Register

Reset: 0000 0000h. Note: this register is an extension of [The MCA NB Configuration Register] F3x44.

Bits	Description
31:27	Reserved.
26	<b>ChgUcToCeEn: change uncorrectable error to correctable error enable.</b> Revision C and earlier revisions: Reserved. Revision D and later: Read-write. 1=The status of uncorrectable errors is changed to appear as correctable errors; F3x4C[UC, PCC] are cleared and a machine check exception will not be raised. For uncorrectable ECC errors, F3x4C[UECC] is cleared and F3x4C[CECC] is set. Note: This field is intended for debug observability.
25	<b>EccSymbolSize: ECC symbol size and code selection.</b> Revision C and earlier revisions: Reserved. Revision D and later: Read-write. 1=x8 symbol size and code used. 0=x4 symbol size and code used. For recommendations on setting, see 2.12.2 [DRAM Considerations for ECC].
24	<b>McaLogErrAddrWdtErr: log error address on WDT errors.</b> Revision C and earlier revisions: Reserved. Revision D and later: Read-write. 1=When a watchdog timeout error occurs (see F3x40[WDTRptEn]), the associated address is logged and F3x4C[AddrV] is set. 0=When a watchdog timeout error occurs, NB state information is saved and F3x4C[AddrV] is cleared. See F3x50 for details on saved information.
22	<b>SyncFloodOnTblWalkErr: sync flood on table walk error enable.</b> Read-write. 1=A sync flood is generated when the DEV or GART table walkers encounter an uncorrectable error. A machine check exception is generated independent of the state of this bit. It is recommended that this bit be set for normal operation.
21	<b>SyncFloodOnCpuLeakErr: sync flood on CPU leak error enable.</b> Read-write. 1=A sync flood is generated when one of the cores encounters an uncorrectable error which cannot be contained to the process on the core. It is recommended that this bit be set for normal operation.
20	<b>SyncFloodOnL3LeakErr: sync flood on L3 cache leak error enable.</b> Read-write. 1=A sync flood is generated when the L3 cache encounters an uncorrectable error which cannot be contained to the process on one core. It is recommended that this bit be set for normal operation.
19:10	Reserved.
9	<b>SyncOnUncNbAryEn: sync flood on uncorrectable NB array error enable.</b> Read-write. 1=Enables sync flood on detection of an error in a NB array that is uncorrectable. BIOS should set this bit to 1 for normal operation.
8	<b>SyncOnProtEn: sync flood on protocol error enable.</b> Read-write. 1=Enables sync flood on detection of a protocol error on a link or in the L3. BIOS should set this bit to 1 for normal operation.
7	<b>SyncFloodOnTgtAbtErr.</b> Read-write. 1=Enable sync flood on generated or received link responses that indicate target aborts. BIOS should set this bit to 1 for normal operation.
6	<b>SyncFloodOnDatErr.</b> Read-write. 1=Enable sync flood on generated or received link responses that indicate data error. BIOS should set this bit to 1 for normal operation.



5	<b>DisPciCfgCpuMstAbtRsp.</b> Read-write. 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>MstAbtChgToNoErrs.</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 MstAbtChgToNoErrs and <a href="#">F3x44[IoMstAbortDis]</a> are both set, MstAbtChgToNoErrs takes precedence.
3	<b>DatErrChgToTgtAbt.</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="#">F3x44[WDTCntSel]</a> .
1	<b>SyncFloodOnUsPwDataErr: sync flood on upstream posted write data error.</b> Read-write. 1=Enable sync flood generation when an upstream posted write data error is detected. BIOS should set this bit to 1 for normal operation.
0	<b>McaLogUsPwDataErrEn: MCA log of upstream posted write data error enable.</b> Read-write. 1=Enable logging of upstream posted write data errors in MCA (if NB MCA registers are appropriately enabled and configured).

### **F3x188 NB Extended Configuration Low Register**

Revision C and earlier revisions: Reset: 0000 0000h. Revision D and later revisions: Reset: 0800 0000h. It is expected that the fields of this register are programmed to the same value in all nodes.

Bits	Description
31:5	Reserved.
4	<b>EnStpGntOnFlushMaskWakeup: enable stop grant on flush mask wakeup.</b> Revision B: Reserved. Revision C and later: Read-write. 1=When in the C1E state, cores that have flushed their L1 and L2 caches are not ramped up for probes. 0=When in the C1E state, cores are ramped based on <a href="#">F3x[84:80][CpuPrbEn]</a> regardless of the state of their caches. See <a href="#">2.4.3.5 [C5]</a> . BIOS should set this bit for revision (DA-C    C3    D1 and later G34    D1 and later C32    E). See <a href="#">2.4.3.3.5 [BIOS Requirements to Initialize Message Triggered C1E]</a> and <a href="#">2.4.2.1.1 [Core Performance Boost (CPB)]</a> .
3:0	Reserved.

### **F3x190 Downcore Control Register**

Cold reset: 0000 0000h. See [2.9.2 \[CPU Cores and Downcore\]](#). Changes to this register do not take effect until after a warm reset.

Bits	Description
31:6	Reserved.
5:0	<b>DisCore[5:0].</b> Read-write. 1=Disable the core. 0=Enable the core. DisCore[5:4] is reserved for revision C and earlier.

### **F3x1A0 L3 Buffer Count Register**

Updates to this register do not take effect until after a warm reset.

- To ensure deadlock free operation the following minimum buffer allocations are required:



CpuCmdBufCnt >= 1

- If the processor includes an L3 cache (as specified by CPUID Fn8000\_0006\_EDX[L3Size]), then to ensure deadlock free operation the following minimum buffer allocations are required:

L3ToSriReqCBC >= 2    L3ToSriReqCBC >= (number of enabled cores)

If the processor does not include an L3 cache, then L3ToSriReqCBC may be 0h.

Bits	Description																
31:15	Reserved.																
14:12	<b>L3ToSriReqCBC: L3 cache to SRI request command buffer count.</b> Read-write. Cold reset: 4h.																
11:9	Reserved.																
8:4	<p><b>L3FreeListCBC: L3 free list command buffer counter for CPU requests.</b> Read-write. Cold reset varies based on number of enabled cores in the node as follows:</p> <table border="1"> <thead> <tr> <th>Cores</th> <th>Cold reset value</th> <th>Cores</th> <th>Cold reset value</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1Ch</td> <td>2</td> <td>18h</td> </tr> <tr> <td>3</td> <td>14h</td> <td>4</td> <td>10h</td> </tr> <tr> <td>5</td> <td>11h</td> <td>6</td> <td>0Eh</td> </tr> </tbody> </table> <p>BIOS should set this field up per the following equation (although lower values are legal):  <math>L3FreeListCBC = 32 - \text{CpuCmdBufCnt} * (\text{number of enabled cores}).</math></p>	Cores	Cold reset value	Cores	Cold reset value	1	1Ch	2	18h	3	14h	4	10h	5	11h	6	0Eh
Cores	Cold reset value	Cores	Cold reset value														
1	1Ch	2	18h														
3	14h	4	10h														
5	11h	6	0Eh														
3	Reserved.																
2:0	<b>CpuCmdBufCnt: CPU to SRI command buffer count.</b> Read-write. Cold reset = 4 if the product includes an L3 cache and the number of enabled cores in the node <= 4; 3 if the product includes an L3 cache and the number of enabled cores in the node >= 5; or 2 if the product does not include an L3 cache. Each core is allocated the number of buffers specified by this field.																

### F3x1C4 L3 Cache Parameter Register

Reset: 0000 0XXXh. This register is reserved if the processor does not include an L3 cache.

Bits	Description
31	<b>L3TagInit: L3 tag initialization.</b> Read; write 1 only. 1=Initialize the L3 cache tag arrays. 0=L3 cache tag initialization is complete. This bit is cleared by hardware when the tag initialization is complete. This bit should not be written while initialization is in progress.
30:14	Reserved.
13:12	<p><b>L3SubcacheSize3: L3 subcache size 3.</b> Revision B: Reserved. Revision C and later revisions: Read-only. Specifies the size of L3 subcache 3.</p> <p>00b=2 MB.  01b=1 MB.  10b=1 MB.  11b=0 MB.</p>
11:10	Reserved.
9:8	<p><b>L3SubcacheSize2: L3 subcache size 2.</b> Revision B: Reserved. Revision C and later revisions: Read-only. Specifies the size of L3 subcache 2.</p> <p>00b=2 MB.  01b=1 MB.  10b=1 MB.  11b=0 MB.</p>
7:5	Reserved.

4	<b>L3SubcacheSize1: L3 subcache size 1.</b> Read-only. Specifies the size of L3 subcache 1. 0=1 MB. 1=0 MB.
3:1	Reserved.
0	<b>L3SubcacheSize0: L3 subcache size 0.</b> Read-only. Specifies the size of L3 subcache 0. 0=1 MB. 1=0 MB.

### F3x1CC IBS Control Register

Reset: 0000 0000h. This register can also be read from [MSRC001\\_103A](#). The BIOS should program this register to enable performance modeling software to use IBS interrupts.

Bits	Description
31:9	Reserved.
8	<b>LvtOffsetVal: local vector table offset valid.</b> Read-write. 1=The offset in LvtOffset is valid. 0=The offset in LvtOffset is not valid and IBS interrupt generation is disabled. The BIOS should set this bit to 1b.
7:4	Reserved.
3:0	<b>LvtOffset: local vector table offset.</b> Read-write. This specifies the address of the IBS LVT entry in the APIC registers as follows: LVT address = (LvtOff shifted left 4 bits) + 500h (see <a href="#">APIC[530:500]</a> ). Only values of 0h-3h may be programmed in this field.

### F3x1D4 Probe Filter Control Register

Revisions C and earlier revisions: Reserved. Revision D and later: Reset: 0000 0000h. See [2.6.5.1 \[Probe Filter\]](#) for details about probe filter operation and requirements.

Bits	Description
31:30	Reserved.
29	<b>PFLoIndexHashEn: probe filter low index hash enable.</b> Read-write. 1=Include address bits[8:12] in the subcache selection hash. 0=Do not include address bits[8:12] in the subcache selection hash. The subcache selection hash is computed using an exclusive OR of system address bits. BIOS should program this to 1.
28	<b>PF EccError: probe filter ECC error.</b> Set-by-hardware; write-1-to-clear. 1=An uncorrectable ECC error was encountered on a directory read.
27:24	<b>PF ErrIntLvtOff: probe filter error interrupt LVT offset.</b> Read-write. This specifies the address of the LVT entry in the APIC registers as follows: LVT address = (LvtOffset shifted left 4 bits) + 500h (see <a href="#">APIC[530:500]</a> ).
23:22	<b>PF ErrInt: probe filter error interrupt type.</b> Read-write. 00b=No interrupt. 01b=APIC LVT based interrupt. 10b=SMI. 11b=Reserved.

21:20	<b>PFPreferredSOREpl: PF preferredSO replacement mode.</b> Read-write. Preferentially retain entries for cache lines shared by multiple nodes and replace entries for cache lines in a single node. 00b=Use preferredSO replacement 14/16 times. 01b=Reserved. 10b=Use preferredSO replacement 12/16 times. 11b=Disable preferredSO replacement. BIOS should program this to 10b.
19	<b>PFInitDone: probe filter initialization done.</b> Read-only; updated-by-hardware. 1=Probe filter initialization is complete. 0=Probe filter is not enabled or initialization is in progress.
18:16	Reserved
15:12	<b>PFSubCacheEn: probe filter subcache enable.</b> Read-write. Each bit of this field applies to one of the up-to-four subcaches of the L3. Bit 0 applies to subcache 0; bit 1 applies to subcache 1; etc. 1=Enable allocation of the probe filter directory within the subcache. BIOS should program this to Fh.
11:10	<b>PFSubCacheSize3: probe filter subcache 3 size.</b> See PFSubCacheSize0.
9:8	<b>PFSubCacheSize2: probe filter subcache 2 size.</b> See PFSubCacheSize0.
7:6	<b>PFSubCacheSize1: probe filter subcache 1 size.</b> See PFSubCacheSize0.
5:4	<b>PFSubCacheSize0: probe filter subcache 0 size.</b> Read-write. This specifies the size of the selected L3 subcache. BIOS should program this to 00b. 00b=1 MB. 01b=2 MB. 1Xb=Reserved.
3:2	<b>PFWayNum: probe filter way number.</b> Read-write. This specifies the number of ways in selected L3 subcache(s) allocated for the probe filter directory. BIOS should program this to 10b. 00b=One way. 01b=Two way. 10b=Four way. 11b=Reserved.
1:0	<b>PFMode: probe filter mode.</b> Read-write. 00b=Probe filter disabled. 01b=Reserved 10b=Probe filter enabled. See 2.6.5.1 [Probe Filter] for BIOS recommendation. 11b=Reserved.

### F3x1E4 SBI Control Register

This register specifies the behavior associated with the SIC and SID pins which may be used to support SMBus-based sideband interface (SBI) protocol. See 2.12.3 [Sideband Interface (SBI)]. F3x1E4 should only be programmed on internal node 0 (F3xE8[IntNodeNum]=00b) if F3xE8[MultiNodeCpu]=1.

Bits	Description
31	<b>SbiRegWrDn: SBI register write done.</b> Reset 1b. Read-only; updated-by-hardware. 1=Write to the SBI registers through F3x1EC has completed. 0=Write to the SBI registers in progress.
30:12	Reserved.

11:8	<b>LvtOffset: local vector table offset.</b> Revision B: Reserved. Revision C: Cold reset: 0000b. Read-write. This 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]).
7	Reserved.
6:4	<b>SbiAddr: SMBus-based sideband interface address.</b> Read-write. Cold reset: For internal node 0, specified by the SA[2:0] strap pins (value matches the pins until the deassertion of RESET_L for a cold reset only; value is not changed by a warm reset); 000b for internal node 1 or products that do not include SA[2:0] pins. Specifies bits[3:1] of the SMBus address of the processor SBI ports. SMBus address bits [3:1] = {~SA[2],SA[1:0]}. SbiAddr must be unique for all processors on the same SMBus segment. It is recommended that BIOS program SbiAddr on internal node 1 to the same value as internal node 0 of the processor.
3	<b>SbTsiDis: SMBus-based sideband temperature sensor interface disable.</b> Read-only. 1=The processor does not support SMBus-based SB-TSI protocol.
2	Reserved.
1	<b>SbRmiDis: SMBus-based sideband remote management interface disable.</b> Read-only. 1=The processor does not support SMBus-based SB-RMI protocol.
0	Reserved.

### F3x1E8 SBI Address Register

Cold reset 0000 0000h. The SB-RMI and SB-TSI registers can be directly accessed by the processor using F3x1E8 and F3x1EC. A read or write access to F3x1EC triggers a read or write to the register specified by F3x1E8[SbiBankSel and SbiRegAddr]. F3x1E8 should only be programmed on internal node 0 (F3xE8[IntNodeNum]=00b) if F3xE8[MultiNodeCpu]=1.

Bits	Description
31:	Reserved.
10:9	<b>SbiByteCnt: SBI byte count.</b> Revision C and earlier revisions: Reserved. Revision D and later: Read-write. This field specifies the number consecutive SBI registers to read or write using F3x1EC. 00b = 1 register. 01b = 2 registers. 10b = 3 registers. 11b = 4 registers.
8	<b>SbiBankSel: SBI register bank select.</b> Revision C and earlier revisions: Reserved. Revision D and later: Read-write. This bit specifies if the SB-TSI or SB-RMI registers are accessed. 1=SB-RMI registers. 0=SB-TSI registers.
7:0	<b>SbiRegAddr: SBI SMBus register address.</b> Read-write. This field specifies the 8-bit address of the SB-TSI or SB-RMI register to access.

### F3x1EC SBI Data Register

Reset 0000 0000h. F3x1EC should only be programmed on internal node 0 (F3xE8[IntNodeNum]=00b) if F3xE8[MultiNodeCpu]=1.

Bits	Description
31:24	<b>SbiRegDat3: SBI SMBus register 3 data.</b> Revision C and earlier revisions: Reserved. Revision D: Read-write. This field specifies the data to be read or written to the SBI register selected by <a href="#">F3x1E8[SbiRegAddr+3]</a> .
23:16	<b>SbiRegDat2: SBI SMBus register 2 data.</b> Revision C and earlier revisions: Reserved. Revision D: Read-write. This field specifies the data to be read or written to the SBI register selected by <a href="#">F3x1E8[SbiRegAddr+2]</a> .
15:8	<b>SbiRegDat1: SBI SMBus register 1 data.</b> Revision C and earlier revisions: Reserved. Revision D: Read-write. This field specifies the data to be read or written to the SBI register selected by <a href="#">F3x1E8[SbiRegAddr+1]</a> .
7:0	<b>SbiRegDat0: SBI SMBus register 0 data.</b> Read-write. This field specifies the data to be read or written to the SBI register selected by <a href="#">F3x1E8[SbiRegAddr]</a> .

### **F3x1EC\_x[FF:01] SB-TSI Registers**

The SB-TSI registers can be accessed by programming [F3x1E8\[SbiRegAddr\]](#) with the offset value and setting [F3x1E8\[SbiBankSel\]=0](#). Accesses to reserved offsets result in undefined behavior. See [2.12.3 \[Sideband Interface \(SBI\)\]](#). The following is the SB-TSI register list:

Offset	Register
01h	CPU Temperature High Byte Register
02h	SB-TSI Status Register
03h	SB-TSI Configuration Register
04h	Update Rate Register
05h-06h	Reserved
07h	High Temperature Threshold High Byte Register
08h	Low Temperature Threshold High Byte Register
09h	SB-TSI Configuration Register
10h	CPU Temperature Low Byte Register
11h	CPU Temperature Offset High Byte Register
12h	CPU Temperature Offset Low Byte Register
13h	High Temperature Threshold Low Byte Register
14h	Low Temperature Threshold Low Byte Register
15h-21h	Reserved
22h	Timeout Configuration Register
23h-6Eh	Reserved
32h	Alert Threshold Register
70h-BEh	Reserved
BFh	Alert Configuration Register
C0h-FDh	Reserved
FEh	Manufacture ID Register
FFh	SB-TSI Revision Register

Bits	Description
31:8	Reserved.
7:0	See the <i>SBI Temperature Sensor Interface (SB-TSI) Specification</i> for register definition.

### F3x1EC\_x1[7F:00] SB-RMI Registers

The SB-RMI registers can be accessed by programming F3x1E8[SbiRegAddr] with the offset value and setting F3x1E8[SbiBankSel]=1. Accesses to reserved offsets result in undefined behavior. See 2.12.3 [Sideband Interface (SBI)]. The following is the SB-RMI register list:

Offset	Register
00h	Interface Revision Register
01h	Control Register
02h	Status Register
03h	Read Size Register
04h	Core Enable Status Register 0
05h	Core Enable Status Register 1
06h	APIC Spin Loop Status Register 0
07h	APIC Spin Loop Status Register 1
08h-0Fh	Reserved
10h-1Fh	Alert Status Registers
20h-2Fh	Alert Mask Registers
30h-37h	Outbound Message Registers
38h-3Fh	Inbound Message Registers
40h	Software Interrupt Register
Revision C:	
41h-FFh	Reserved
Revision D and later:	
41h	RMI Core Number
42h-FFh	Reserved

Bits	Description
31:8	Reserved.
7:0	See the <i>Advanced Platform Management Link (APML) Specification</i> for register definition.

### F3x1F0 Product Information Register

Bits	Description
31:26	Reserved.
25:20	<b>MaxSwPstateCpuCof: maximum software P-state core COF.</b> Revision D and earlier: Reserved. Revision E: Read-only. Cold reset value varies by product. 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 2.4.2.1.2.1 [Software P-state Numbering].
18:16	<b>NbPstate: Northbridge P-state.</b> Revision C2 and earlier and revision D and later: Reserved. The default value of MSRC001_00[68:64][NbDid] specifies NB P-state support. Revision C3: Read-only. If NbPstate=0h, no action is required by BIOS and NB P-states are not supported. If NbPstate != 0h, this field specifies the highest performance P-state that enables NB P-state 1. BIOS should program MSRC001_00[68:64][NbDid]=1 and MSRC001_00[68:64][NbVid]=F4x1F4[NbVid1] for the P-state specified by NbPstate and all lower performance P-states where MSRC001_00[68:64][PstateEn]=1. See 2.4.2.6 [BIOS Requirements for P-State Initialization and Transitions] and 2.4.2.2 [NB P-states].
15:0	<b>BrandId.</b> Read-only. Brand identifier. This is identical to CPUID Fn8000_0001_EBX[BrandId].

### F3x1FC Product Information Register

Revision C and earlier revisions:

Bits	Description
31	<b>DualPlaneOnly</b> . Revision B: Reserved. Revision C: Read-only. Reset: value varies by product. Specifies the infrastructure that supports the processor. 0=The processor is supported by both the single- and dual-plane infrastructures. 1=The processor is only supported by the dual-plane infrastructure. See 2.4.2.7 [BIOS Configuration for Dual-plane Only Support].
30:26	Reserved.
25:22	<b>SinglePlaneNbIdd[3:0]</b> . Read-only. Specifies the NbIdd value for platforms with unified VDD and VDDNB power planes. The NbIdd is specified in amps according to the following formula: $NbIdd = SinglePlaneNbIdd[3:0] * 2$ .
21:17	<b>DualPlaneNbVidOff[4:0]</b> . Read-only. Specifies the NbVid offset value required for NB operation at the frequency specified by DualPlaneNbFid. DualPlaneNbVidOff is applied relative to SinglePlaneNbVid using the following formula: $DualPlaneNbVid = SinglePlaneNbVid - \{00b, DualPlaneNbVidOff[4:0]\}$ . See 2.4.2.9 [BIOS Northbridge COF and VID Configuration].
16:14	<b>DualPlaneNbFidOff[2:0]</b> . Read-only. Specifies the NbFid value for platforms with separate VDD and VDDNB power planes. This offset is applied relative to SinglePlaneNbFid using the following formula: $DualPlaneNbFid = SinglePlaneNbFid + \{00b, DualPlaneNbFidOff[2:0]\}$ . See 2.4.2.9 [BIOS Northbridge COF and VID Configuration].
13:7	<b>SinglePlaneNbVid[6:0]</b> . Read-only. Specifies the NbVid value required for NB operation at the frequency specified by SinglePlaneNbFid. See 2.4.2.9 [BIOS Northbridge COF and VID Configuration].
6:2	<b>SinglePlaneNbFid[4:0]</b> . Read-only. Specifies the NbFid value for platforms with unified VDD and VDDNB power planes. See 2.4.2.9 [BIOS Northbridge COF and VID Configuration].
1	<b>NbVidUpdateAll</b> . Read-only. Indicates that software is required to update NbVid after cold reset based on the sequence defined in section 2.4.2.9 [BIOS Northbridge COF and VID Configuration].
0	<b>NbCofVidUpdate</b> . Read-only. Indicates that software is required to update the NB COF and NbVid after cold reset based on the sequence defined in section 2.4.2.9 [BIOS Northbridge COF and VID Configuration].

Revision D and later:

Bits	Description
31	<b>DualPlaneOnly</b> . Revision D: Reserved. Revision E: Read-only. Reset: value varies by product. Specifies the infrastructure that supports the processor. 0=The processor is supported by both the single- and dual-plane infrastructures. 1=The processor is only supported by the dual-plane infrastructure. See 2.4.2.7 [BIOS Configuration for Dual-plane Only Support].
30:0	Reserved.



### 3.7 Function 4 Link Control Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention. See 2.11 [Configuration Space] for details about how to access this space.

#### F4x00 Device/Vendor ID Register

Reset: 1204 1022h.

Bits	Description
31:16	<b>DeviceID: device ID.</b> Read-only.
15:0	<b>VendorID: vendor ID.</b> Read-only.

#### F4x04 Status/Command Register

Reset: 00?0 0000h.

Bits	Description
31:16	<b>Status.</b> Read-only. Only bit[20] 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.

#### F4x08 Class Code/Revision ID Register

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.

#### F4x0C Header Type Register

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.

#### F4x34 Capabilities Pointer Register

Reset: 0000 00??h.

Bits	Description
31:8	Reserved.
7:0	<b>CapPtr: capabilities pointer.</b> Read-only. Specifies the offset of the link capabilities block based on which links are supported and ungangled. The value provided is: <ul style="list-style-type: none"> <li>80h If link 0 is supported and ungangled.</li> <li>A0h If link 0 is gangled/unsupported and link 1 is supported and ungangled.</li> <li>C0h If link 0 and 1 are gangled/unsupported and link 2 is supported and ungangled.</li> <li>E0h If link 0, 1, and 2 are gangled/unsupported and link 3 is supported and ungangled.</li> </ul>

**F4x[E0, C0, A0, 80] Sublink 1 Capability Registers**

See F0x[E0, C0, A0, 80] for details of this register. If the link is ganged or not supported, then this register is reserved. Note: the CapPtr field is controlled similarly to F0x[E0, C0, A0, 80][CapPtr]. However, based on whether the next link is supported *and* unganged.

**F4x[E4, C4, A4, 84] Sublink 1 Control Registers**

See F0x[E4, C4, A4, 84] for details of this register. If the link is ganged or not supported, then this register is reserved.

**F4x[E8, C8, A8, 88] Sublink 1 Frequency/Revision Registers**

See F0x[E8, C8, A8, 88] for details of this register. If the link is ganged or not supported, then this register is reserved.

**F4x[EC, CC, AC, 8C] Sublink 1 Feature Capability Registers**

See F0x[EC, CC, AC, 8C] for details of this register. If the link is ganged or not supported, then this register is reserved.

**F4x[F0, D0, B0, 90] Sublink 1 Base Channel Buffer Count Registers**

See F0x[F0, D0, B0, 90] for details of this register. If the link is ganged or not supported, then this register is reserved.

**F4x[F4, D4, B4, 94] Sublink 1 Isochronous Channel Buffer Count Registers**

See F0x[F4, D4, B4, 94] for details of this register. If the link is ganged or not supported, then this register is reserved.

**F4x[F8, D8, B8, 98] Sublink 1 Link Type Registers**

See F0x[F8, D8, B8, 98] for details of this register. If the link is ganged or not supported, then this register is reserved.

**F4x[FC, DC, BC, 9C] Sublink 1 Link Frequency Extension Registers**

Revision C and earlier: Reserved. Revision D and later: See F0x[FC, DC, BC, 9C] for details of this register. If the link is ganged or not supported, then this register is reserved.

**F4x15C Core Performance Boost Control**

Revision D and earlier: Reserved. Revision E: See 2.4.2.1.1 [Core Performance Boost (CPB)].

Bits	Description
31	<p><b>BoostLock.</b> Read-only. Reset: value varies by product. 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>• F4x15C[NumBoostStates].</li> <li>• F4x16C[CstateCnt].</li> </ul>

30:3	Reserved.										
2	<b>NumBoostStates: number of boosted states.</b> If <a href="#">F4x15C[BoostLock]=1</a> , read-only; else read-write. Reset: value varies by product. Specifies the number of P-states that are considered boosted P-states. See <a href="#">2.4.2.1.1 [Core Performance Boost (CPB)]</a> for details.										
1:0	<b>BoostSrc: boost source.</b> Read-write. Reset: 0. Specifies whether CPB is enabled or disabled. BIOS should program this to 11b if <a href="#">CPUID Fn8000_0007[CPB]=1</a> . <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>Reserved.</td> </tr> <tr> <td>10b</td> <td>Reserved.</td> </tr> <tr> <td>11b</td> <td>Boosting enabled.</td> </tr> </tbody> </table>	Bits	Description	00b	Boosting disabled.	01b	Reserved.	10b	Reserved.	11b	Boosting enabled.
Bits	Description										
00b	Boosting disabled.										
01b	Reserved.										
10b	Reserved.										
11b	Boosting enabled.										

### F4x16C Cstate Boost Control

Revision D and earlier: Reserved. Revision E: Reset values vary by product.

Bits	Description
31:12	Reserved.
11:9	<b>CstateCnt: C-state count.</b> If <a href="#">F4x15C[BoostLock]</a> then read-only; else read-write. Reset value varies by product. Specifies the number of cores that must be in the C1 state before a transition can occur to a boosted P-state. See <a href="#">2.4.2.1.1 [Core Performance Boost (CPB)]</a> . A value of 0 disables access to a boosted P-state.
8:0	Reserved.

### F4x1[98, 90, 88, 80] Link Phy Offset Registers

Cold reset: 8000 0000h. The links each include an array of registers called [F4x1\[9C, 94, 8C, 84\]\\_x\[NN:00\]](#), which are defined following [F4x1\[9C, 94, 8C, 84\]](#). These are used primarily to control link electrical parameters and to program the link BIST engine. [\[The Link Phy Offset Registers\] F4x1\[98, 90, 88, 80\]](#) and [\[The Link Phy Data Port\] F4x1\[9C, 94, 8C, 84\]](#) are used to access these registers. The register number (i.e., the number that follows “\_x” in the register mnemonic) is specified by [F4x1\[98, 90, 88, 80\]\[LinkPhyOffset\]](#). Access to these registers is accomplished as follows:

- Reads:
  - Write the register number to [F4x1\[98, 90, 88, 80\]\[LinkPhyOffset\]](#) with [F4x1\[98, 90, 88, 80\]\[LinkPhyWrite\]=0](#).
  - Poll [F4x1\[98, 90, 88, 80\]\[LinkPhyDone\]](#) until it is high.
  - Read the register contents from [F4x1\[9C, 94, 8C, 84\]](#).
- Writes:
  - Write all 32 bits of register data to [F4x1\[9C, 94, 8C, 84\]](#) (individual byte writes are not supported).
  - Write the register number to [F4x1\[98, 90, 88, 80\]\[LinkPhyOffset\]](#) with [F4x1\[98, 90, 88, 80\]\[LinkPhyWrite\]=1](#).
  - Poll [F4x1\[98, 90, 88, 80\]\[LinkPhyDone\]](#) until it is high to ensure that the contents of the write have been delivered to the phy.

The links also include an array of direct map registers. A link phy register is not a direct map register unless it is specified in the register description. The read and write access to the direct map registers is similar to the process described above except for the following:

- F4x1[98, 90, 88, 80][DirectMapEn] must be set.
- The register number (i.e., the number that follows “\_x” in the register mnemonic) expands to 16 bit wide and is specified by F4x1[98, 90, 88, 80][{UpperLinkPhyOffset, LinkPhyOffset}]. For example, to access [The DLL Control and Test Register 3] F4x1[9C, 94, 8C, 84]\_x[530A, 520A], F4x1[98, 90, 88, 80][{UpperLinkPhyOffset, LinkPhyOffset}] must be programmed as 530Ah or 520Ah.

Note: Read or write accesses to undocumented or undefined register numbers can result in undefined behavior.

Note: Read or write accesses to links that are not implemented on the package complete with undefined results. See 2.7.1.3.1 [Unused Links].

F4x180 and F4x184 are for link 0; F4x188 and F4x18C are for link 1; F4x190 and F4x194 are for link 2; F4x198 and F4x19C are for link 3.

Note: Each link may only have one outstanding read or write at a time.

Bits	Description
31	<b>LinkPhyDone: link phy access done.</b> Read-only. 1=The access to one of the F4x1[9C, 94, 8C, 84]_x[NN:00] registers is complete. 0=The access is still in progress.
30	<b>LinkPhyWrite: link phy read/write select.</b> Read-write. 0=Read one of the F4x1[9C, 94, 8C, 84]_x[NN:00] registers. 1=Write one of the F4x1[9C, 94, 8C, 84]_x[NN:00] registers.
29	<b>DirectMapEn: direct map enable.</b> Read-write. Cold reset 0. 1=Enable link phy address direct map mode. This bit should only be set to access direct map link phy address registers as specified in the register descriptions.
28:16	Reserved.
15:10	<b>UpperLinkPhyOffset: upper link phy offset address bits.</b> Read-write.
9:0	<b>LinkPhyOffset: link phy offset.</b> Read-write.

### **F4x1[9C, 94, 8C, 84] Link Phy Data Port**

See F4x1[98, 90, 88, 80] for details about this port.

### **F4x1[9C, 94, 8C, 84]\_x[D0, C0] Link Phy Impedance Registers**

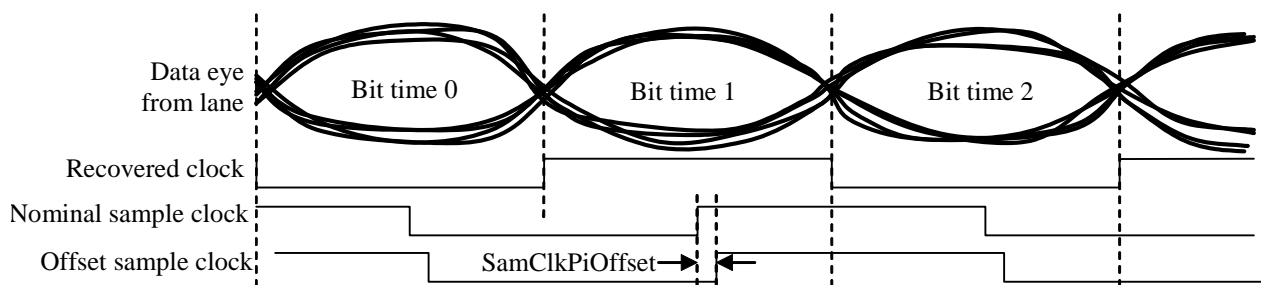
See F4x1[98, 90, 88, 80] for register access information. The \_xC0 register number specifies values for CAD[7:0], CTL0, and CLK0; the \_xD0 register number specifies values for CAD[15:8], CTL1, and CLK1. These register bits are updated as specified by F0x16C[ImmUpdate]. Note: updates to these registers that result in a change to impedance may not take effect in the phy for up to 2 microseconds after the update to this register completes (or until a disconnect if ImmUpdate is clear).

Bits	Description												
31:29	<p><b>RttCtl: receiver termination resistance (Rtt) control.</b> Read-write. Cold reset: 0. This field specifies how the receiver termination resistance value is calculated. All values between 00h and 1Fh are valid.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Rtt is as determined by the compensation circuit, F4x1[9C, 94, 8C, 84]_xE0[RttRawCal].</td> </tr> <tr> <td>001b</td> <td>Rtt is as specified by the index field (F4x1[9C, 94, 8C, 84]_x[D0, C0][RttIndex]).</td> </tr> <tr> <td>010b</td> <td>Rtt is as specified by the difference: RttRawCal - RttIndex. If this results in a value that is less than 00h, then 00h is used. 011bRtt is as specified by the sum: RttRawCal + RttIndex. If this results in a value that is greater than 1Fh, then 1Fh is used.</td> </tr> <tr> <td>100b</td> <td>Enable only one tap of the Rtt resistor, as specified by RttIndex, and disable the base resistor that is normally always enabled. This is intended for testing purposes only.</td> </tr> <tr> <td>101b - 111b:</td> <td>reserved.</td> </tr> </tbody> </table> <p>For all modes (except 100b), higher values reduce the resistance of Rtt and lower values increase the resistance of Rtt. See 2.7.2 [Termination and Compensation] for more information about compensation.</p> <p>Revision C and later revisions: If RttCtl is programmed to either 011b or 100b, the value of RttRawCal + RttIndex must be less than or equal to 24.</p>	Bits	Definition	000b	Rtt is as determined by the compensation circuit, F4x1[9C, 94, 8C, 84]_xE0[RttRawCal].	001b	Rtt is as specified by the index field (F4x1[9C, 94, 8C, 84]_x[D0, C0][RttIndex]).	010b	Rtt is as specified by the difference: RttRawCal - RttIndex. If this results in a value that is less than 00h, then 00h is used. 011bRtt is as specified by the sum: RttRawCal + RttIndex. If this results in a value that is greater than 1Fh, then 1Fh is used.	100b	Enable only one tap of the Rtt resistor, as specified by RttIndex, and disable the base resistor that is normally always enabled. This is intended for testing purposes only.	101b - 111b:	reserved.
Bits	Definition												
000b	Rtt is as determined by the compensation circuit, F4x1[9C, 94, 8C, 84]_xE0[RttRawCal].												
001b	Rtt is as specified by the index field (F4x1[9C, 94, 8C, 84]_x[D0, C0][RttIndex]).												
010b	Rtt is as specified by the difference: RttRawCal - RttIndex. If this results in a value that is less than 00h, then 00h is used. 011bRtt is as specified by the sum: RttRawCal + RttIndex. If this results in a value that is greater than 1Fh, then 1Fh is used.												
100b	Enable only one tap of the Rtt resistor, as specified by RttIndex, and disable the base resistor that is normally always enabled. This is intended for testing purposes only.												
101b - 111b:	reserved.												
28:21	Reserved.												
20:16	<p><b>RttIndex: receiver termination resistance (Rtt) index.</b> Read-write. Revision B: Cold reset: 1_1111b. Revision C and later revisions: Cold reset: 0_0000b. See RttCtl for details about how this field is used.</p>												
15:13	<p><b>RonCtl: transmitter resistance (Ron) control.</b> Read-write. Cold reset: 0. This field specifies how the transmitter resistance value is calculated.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Ron is as determined by the compensation circuit, F4x1[9C, 94, 8C, 84]_xE0[RonRawCal].</td> </tr> <tr> <td>001b</td> <td>Ron is as specified by the index field (F4x1[9C, 94, 8C, 84]_x[D0, C0][RonIndex]).</td> </tr> <tr> <td>010b</td> <td>Ron is as specified by the difference: RonRawCal - RonIndex. If this results in a value that is less than 00h, then 00h is used. 011bRon is as specified by the sum: RonRawCal + RonIndex. If this results in a value that is greater than 1Fh, then 1Fh is used.</td> </tr> <tr> <td>100b</td> <td>Enable only one tap of the Ron resistor, as specified by RonIndex, and disable the base resistor that is normally always enabled. This is intended for testing purposes only.</td> </tr> <tr> <td>101b - 111b:</td> <td>reserved.</td> </tr> </tbody> </table> <p>For all modes (except 100b), higher values reduce the resistance of Ron and lower values increase the resistance of Ron. See 2.7.2 [Termination and Compensation] for more information about compensation.</p> <p>Revision C and later: If RonCtl is programmed to either 011b or 100b, the value of RonRawCal + RonIndex must be less than or equal to 23.</p>	Bits	Definition	000b	Ron is as determined by the compensation circuit, F4x1[9C, 94, 8C, 84]_xE0[RonRawCal].	001b	Ron is as specified by the index field (F4x1[9C, 94, 8C, 84]_x[D0, C0][RonIndex]).	010b	Ron is as specified by the difference: RonRawCal - RonIndex. If this results in a value that is less than 00h, then 00h is used. 011bRon is as specified by the sum: RonRawCal + RonIndex. If this results in a value that is greater than 1Fh, then 1Fh is used.	100b	Enable only one tap of the Ron resistor, as specified by RonIndex, and disable the base resistor that is normally always enabled. This is intended for testing purposes only.	101b - 111b:	reserved.
Bits	Definition												
000b	Ron is as determined by the compensation circuit, F4x1[9C, 94, 8C, 84]_xE0[RonRawCal].												
001b	Ron is as specified by the index field (F4x1[9C, 94, 8C, 84]_x[D0, C0][RonIndex]).												
010b	Ron is as specified by the difference: RonRawCal - RonIndex. If this results in a value that is less than 00h, then 00h is used. 011bRon is as specified by the sum: RonRawCal + RonIndex. If this results in a value that is greater than 1Fh, then 1Fh is used.												
100b	Enable only one tap of the Ron resistor, as specified by RonIndex, and disable the base resistor that is normally always enabled. This is intended for testing purposes only.												
101b - 111b:	reserved.												
12:5	Reserved.												
4:0	<p><b>RonIndex: transmitter resistance (Ron) index.</b> Read-write. Revision B: Cold reset: 1_1111b. Revision C and later revisions: Cold reset: 0_0000b. See RonCtl for details about how this field is used.</p>												

### **F4x1[9C, 94, 8C, 84]\_x[D1, C1] Link Phy Receiver Loop Filter Registers**

See F4x1[98, 90, 88, 80] for register access information. The \_xC1 register number specifies values for CAD[7:0], CTL0, and CLK0; the \_xD1 register number specifies values for CAD[15:8], CTL1, and CLK1.

These register bits are updated as specified by [F0x16C\[ImmUpdate\]](#).



**Figure 18: Link phy recovered clock and sample clock.**

When the link is in a mode that relies on dynamic phase alignment (automatic sample-clock correction), then the processor generates a recovered clock for each lane based on transitions in the lane. The ideal recovered clock transitions at exactly the same time as the transitions in the lane. Phase detection logic detects if the recovered clock transitions before or after the lane transition. The digital loop filter (DLF) is logic that adjusts the phase of the recovered clock such that its transitions match the transition time of the lane as much as possible. The DLF counts the number of times the lane transitions before the recovered clock versus after to determine whether to adjust the recovered clock phase. The DLF uses an 8-bit counter, called the loop filter counter (LFC) for this purpose. The LFC controls are included in this register. They specify DLF behavior as follows:

- LfcMax is programmed to be greater than LfcMin.
- The LFC is initialized to LfcMin.
- The LFC is updated periodically. The logic keeps a tally of the number of lane transitions occurring before and after the recovered clock transition within each update period.
- To start, if there is a net lane transition occurs after the recovered clock transition within the update period, the LFC is incremented by the net value; on the other hand, if there is a net lane transition occurs before the recovered clock transition, the LFC is decremented. However, if the LFC is ever decremented while it is zero, these rules are reversed (and the LFC is incremented instead). Thus, if there is a phase correction needed, the LFC trends either upward or downward; if it trends downward, it hits zero and then trends upward again.
- If the LFC reaches LfcMax value, then (1) the phase of the recovered clock is adjusted in the appropriate direction, (2) the LFC is set to the LfcMin value.

The LfcMin and LfcMax fields are designed to improve the stability of the recovered clock phase while improving the response time for multiple phase updates in the same direction. For example, if the recovered clock phase needs several adjustments in the same direction, then: the LFC increments until it hits LfcMax value and then be set to LfcMin (and trigger a phase adjustment); then it would increment to LfcMax value again to trigger the next phase adjustment. If, however, the next phase adjustment needs to be in the opposite direction, the LFC would decrement to zero, change direction, and then increment up to LfcMax again. In this way, phase adjustments in the same direction occur more quickly than phase adjustments in the opposite direction of the prior phase adjustment.

The nominal sample clock is offset by 90 degrees from the *recovered clock*. An offset can be inserted to move the sample clock from the nominal position, based on SamClkPiOffset and SamClkPiOffsetSign.

Bits	Description
31:30	Reserved.

29:22	<b>LfcMax: loop filter counter maximum value.</b> Read-write. Revision B: Cold reset: 80h. Revisions C and later: Cold reset: 20h. Recommended BIOS setting is 20h for higher link frequencies. For lower link frequencies, lower LfcMax value can be programmed to further optimize clock recovery and hence link reconnect time; for example, LfcMax can be 10h when link frequency is 1.2GHz.
21:14	<b>LfcMin: loop filter counter minimum value.</b> Read-write. Revision B: Cold reset: 40h. Revisions C and later: Cold reset: 10h. Recommended BIOS setting is 10h for higher link frequencies. For lower link frequencies, lower LfcMin value can be programmed to further optimize clock recovery and hence link reconnect time; for example, LfcMin can be 08h when link frequency is 1.2GHz.
13:10	Reserved.
9:8	Must be 10b. Read-write.
7	<b>SamClkPiOffsetEn: sample clock phase interpolator offset enable.</b> Read-write. Cold reset: 0. 1=Enable offset insertion around the nominal sample clock position.
6:4	<b>SamClkPiOffset: sample clock phase interpolator offset setting.</b> Read-write. Cold reset: X. This field specifies the magnitude of the offset of the sample clock from the nominal position. See <a href="#">Figure 18</a> . This field is encoded as follows. <ul style="list-style-type: none"> <li>• <b>Sample clock phase interpolator offset</b> = (SamClkPiOffset + 1) * step size.</li> <li>• If link speed is &gt;3.6GT/s, the expected typical step size is 2ps with a +/-1ps error.</li> <li>• If link speed is &lt;=3.6GT/s, the expected typical step size is 3ps with a +/-1ps error.</li> </ul>
3	<b>SamClkPiOffsetSign: sample clock phase interpolator offset setting sign bit.</b> Read-write. Cold reset: X. 0=Sample clock is moved to before the nominal position. 1=Sample clock is moved to after the nominal position. See SamClkPiOffset and <a href="#">Figure 18</a> .
2:0	Reserved.

### **F4x1[9C, 94, 8C, 84]\_x[D3, C3] Link Phy Timing Margin Registers**

See [F4x1\[98, 90, 88, 80\]](#) for register access information. The \_xC3 register number specifies values for CAD[7:0], CTL0, and CLK0; the \_xD3 register number specifies values for CAD[15:8], CTL1, and CLK1.

The built in jitter injection test mode is useful for checking the clock data recovery tracking bandwidth of the receiver. By forcing the sample clock to move from the lock position by a controlled amount and then observing the time it takes to recover, the tracking rate and bandwidth can be estimated. This register provides the control of the test mode.

The jitter injection test mode works as follows.

- The circuit is clocked by a jitter injection clock derived from dividing the link forwarded clock by 2.5; for example, if the link speed is 5.2GT/s and the link forwarded clock frequency is 2.6GHz, the jitter injection clock frequency becomes 1.04GHz.
- There are 2 phases, the on phase and the off phase. It starts with the on phase once the test mode is enabled.
- During the on phase, at every tick of jitter injection clock, the sample clock is moved away from the nominal lock position by  $1/96 * UI$ .
- The direction of adjustment is specified by JitterInjDir.
- The on phase adjustment continues for a number of times as specified by JitterInjOnCnt.
- Then the adjustment turns off for a duration specified by {JitterInjOffCnt, JitterInjOnCnt} \* jitter injection clock period, this is known as the off phase. During this time, clock data recovery resumes to try to adjust the position of the sample clock back to the center of the data eye.
- The off phase is followed by the on phase again. The process continues to alternate between the on phase and the off phase until the jitter injection test mode is disabled.



In addition, the JitterInjHold bit may be set to inject a hold state at the end of the on phase. This stops clock data recovery from resuming after the on phase, hence holding the sample clock at its last adjusted position until the JitterInjHold bit is cleared. This test mode may be useful for margining the width of the input data eye.

Note: This margining mechanism is not characterized for precision jitter adjustments or measurements.

Bits	Description						
31	Reserved.						
30	<b>JitterInjEn: jitter injection enable.</b> Read-write. Cold reset: 0. 1=Jitter injection test mode is enabled.						
29	<b>JitterInjDir: jitter injection direction.</b> Read-write. Cold reset: 0. <table border="1"> <thead> <tr> <th>Bit</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Move clock before the nominal lock position.</td> </tr> <tr> <td>1</td> <td>Move clock after the nominal lock position.</td> </tr> </tbody> </table>	Bit	Definition	0	Move clock before the nominal lock position.	1	Move clock after the nominal lock position.
Bit	Definition						
0	Move clock before the nominal lock position.						
1	Move clock after the nominal lock position.						
28:23	<b>JitterInjOnCnt: jitter injection on count.</b> Read-write. Cold reset: 0.						
22:16	Reserved.						
15:10	<b>JitterInjOffCnt:jitter injection off count.</b> Read-write. Cold reset: 0. The jitter injection off time count is a 12bit code, this field specifies the most significant 6 bits. The least significant 6 bits are the same as JitterInjOnCnt.						
9	<b>JitterInjHold:jitter injection hold.</b> Read-write. Cold reset: 0. 1=Jitter injection hold is enabled.						
8:0	Reserved.						

#### **F4x1[9C, 94, 8C, 84]\_x[D4, C4] Link Phy DFE and DFR Control Registers**

See F4x1[98, 90, 88, 80] for register access information. The \_xC4 register number specifies values for CAD[7:0], CTL0, and CLK0; the \_xD4 register number specifies values for CAD[15:8], CTL1, and CLK1. If there is a requirement to optimize performance on a per-lane basis, see F4x1[9C, 94, 8C, 84]\_x4[9:0][8, 0]6 for more information. These register bits are updated as specified by F0x16C[ImmUpdate].

The processor supports decision feedback restore (DFR), a function that enables on-chip AC coupling on the receiver path in Gen3 DC-coupled mode, to improve the receiver's ability to operate over a longer channel. In this mode, the receiver on the processor must be programmed with the expected peak single-ended DC voltage level over the single-ended DC common mode voltage level, as seen by the receiver, when a static 1 or 0 is driven. For example, without deemphasis at nominal supply voltage of 1.2V, the peak single ended voltage is expected to be 300mV ideally above the single ended DC common mode voltage level. The value is dependent on the deemphasis setting of the transmitter on the other end of the channel.

For revision D and later, decision feedback equalization (DFE) can be enabled to enhance Gen3 link operation. Once enabled, the receiver uses the logic level of the previous data bit to adjust the voltage threshold of the sampler in the direction that causes the sampler to switch sooner when the data bit transitions to the opposite logic level for the next bit. The control and DFE voltage level are included in this register.

Bits	Description
31:16	Reserved.

15:10	<p><b>DCV: transmit single ended DC voltage level.</b> Read-write. Cold reset: 0. This field specifies the peak single-ended DC voltage level over the single-ended DC common mode voltage level, full swing or deemphasized, of the transmitter. BIOS should program this field based on the following equation (rounded to the nearest integer):</p> $DCV = 32 * 10^{(DE/20)}$ <p>where DE is the far-device deemphasis setting.</p> <p>The following is a list of typical settings:</p> <table border="1"> <thead> <tr> <th>Far-device deemphasis setting</th> <th>DCV</th> </tr> </thead> <tbody> <tr> <td>No deemphasis</td> <td>20h</td> </tr> <tr> <td>-3dB postcursor</td> <td>17h</td> </tr> <tr> <td>-6dB postcursor</td> <td>10h</td> </tr> <tr> <td>-8dB postcursor</td> <td>0Dh</td> </tr> </tbody> </table>	Far-device deemphasis setting	DCV	No deemphasis	20h	-3dB postcursor	17h	-6dB postcursor	10h	-8dB postcursor	0Dh
Far-device deemphasis setting	DCV										
No deemphasis	20h										
-3dB postcursor	17h										
-6dB postcursor	10h										
-8dB postcursor	0Dh										
9	<p><b>VdcDacFine.</b> Read-write. Revision B: Reserved. Revision C and later revisions: Cold reset: 0. 1=Changes the resolution of DCV and limits range by half.</p>										
8	Reserved.										
7	<p><b>DfeEn: DFE enable.</b> Revision C and earlier revisions: Reserved. Revision D and later: Read-write. Cold reset: 0. 1=Decision feedback equalization is enabled.</p>										
6:5	<p><b>DfeVoltage: DFE offset voltage level.</b> Revision C and earlier revisions: Reserved. Revision D and later: Read-write. Cold reset: 0. This field specifies the magnitude of the DFE offset voltage.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>DFE offset voltage=25mV.</td> </tr> <tr> <td>01b</td> <td>DFE offset voltage=12.5mV.</td> </tr> <tr> <td>10b</td> <td>DFE offset voltage=20mV.</td> </tr> <tr> <td>11b</td> <td>DFE offset voltage=35mV.</td> </tr> </tbody> </table>	Bits	Definition	00b	DFE offset voltage=25mV.	01b	DFE offset voltage=12.5mV.	10b	DFE offset voltage=20mV.	11b	DFE offset voltage=35mV.
Bits	Definition										
00b	DFE offset voltage=25mV.										
01b	DFE offset voltage=12.5mV.										
10b	DFE offset voltage=20mV.										
11b	DFE offset voltage=35mV.										

#### **F4x1[9C, 94, 8C, 84]\_x[D5, C5] Link Phy Deemphasis Value Registers**

See F4x1[98, 90, 88, 80] for register access information. The \_xC5 register number specifies the deemphasis values for CAD[7:0], CTL0, and CLK0; the \_xD5 register number specifies the deemphasis values for CAD[15:8], CTL1, and CLK1. If there is a requirement to optimize deemphasis settings on a per-lane basis, see F4x1[9C, 94, 8C, 84]\_x6[9:0][8, 0]0 for more information. See 2.7.3 [Equalization] for more information about deemphasis. Links connecting internal nodes as specified by F0x1A0[IntLnkRoute] do not require deemphasis. These register bits are updated as specified by F0x16C[ImmUpdate].

For Gen3 links, the fields in this register can be programmed during link initialization to select the right deemphasis setting shown in Table 101

**Table 101: Gen3 deemphasis settings**

Gen3 deemphasis <sup>1</sup> setting <sup>2</sup>	DL1, DL2, DP1	PostCur1En	PostCur2En	PreCur1En	MapPostCur2En
No deemphasis (0dB)	00h, 00h, 00h	0	0	0	0
-3dB postcursor	12h, 00h, 00h	1	0	0	0
-6dB postcursor	1Fh, 00h, 00h	1	0	0	0

- MapPreCurEn=0 for all the supported Gen 3 deemphasis settings.
- Deemphasis is not supported by the transmit clock lanes.
- This setting is expected to be used for links with very high loss.

**Table 101: Gen3 deemphasis settings**

Gen3 deemphasis <sup>1</sup> setting <sup>2</sup>	DL1, DL2, DP1	PostCur1En	PostCur2En	PreCur1En	MapPostCur2En
-8dB postcursor	1Fh, 06h, 00h	1	1	0	1
-11dB postcursor	1Fh, 0Dh, 00h	1	1	0	1
-11dB postcursor with -8dB precursor <sup>3</sup>	1Fh, 06h, 07h	1	1	1	1

- MapPreCurEn=0 for all the supported Gen 3 deemphasis settings.
- Deemphasis is not supported by the transmit clock lanes.
- This setting is expected to be used for links with very high loss.

Deemphasis is not supported when operating at Gen1 link frequencies. PostCur1En, PostCur2En, and PreCur1En must be 0 for Gen1 links.

Table 102 and Table 103 show recommended deemphasis settings for systems that have not been fully modeled and can also be used as estimates to begin margining work.

**Table 102: Estimated deemphasis (dB) for microstrip routing**

Length of trace (in)	Frequency of operation (GHz)					
	3.2	2.6	2.4	2.2	2.0	1.8
4	-3	0 <sup>1</sup> / <sub>-3</sub> <sup>2</sup> / <sub>-3</sub> <sup>3</sup>	0 <sup>1</sup> / <sub>0</sub> <sup>2</sup> / <sub>-3</sub> <sup>3</sup>	0	0	0
5	-3	-3	0 <sup>1</sup> / <sub>-3</sub> <sup>2</sup> / <sub>-3</sub> <sup>3</sup>	0	0	0
6	-3	-3	-3	0	0	0
7	-3	-3	-3	0 <sup>1</sup> / <sub>-3</sub> <sup>2</sup> / <sub>-3</sub> <sup>3</sup>	0	0
8	-3	-3	-3	-3	0	0
9	-6	-3	-3	-3	0	0
10	-6	-3	-3	-3	0	0
11	-6	-3	-3	-3	-3	0
12	-6	-3	-3	-3	-3	-3
13	-6	-3 <sup>1</sup> / <sub>-3</sub> <sup>2</sup> / <sub>-6</sub> <sup>3</sup>	-3	-3	-3	-3
14	-6	-3 <sup>1</sup> / <sub>-6</sub> <sup>2</sup> / <sub>-6</sub> <sup>3</sup>	-3	-3	-3	-3
15	-6	-6	-3	-3	-3	-3
16	-6	-6	-3 <sup>1</sup> / <sub>-6</sub> <sup>2</sup> / <sub>-6</sub> <sup>3</sup>	-3	-3	-3
17	-6	-6	-6	-3	-3	-3
18	-8	-6	-6	-3	-3	-3
19	-8	-6	-6	-3 <sup>1</sup> / <sub>-3</sub> <sup>2</sup> / <sub>-6</sub> <sup>3</sup>	-3	-3
20	-8	-6	-6	-6	-3	-3

- 0 connectors
- 1 connector
- 2 connectors

**Table 103: Estimated deemphasis (dB) for stripline routing**

Length of trace (in)	Frequency of operation (GHz)					
	3.2	2.6	2.4	2.2	2.0	1.8
4	-3	0 <sup>1</sup> / <sub>-3<sup>2</sup></sub> / <sub>-3<sup>3</sup></sub>	0 <sup>1</sup> / <sub>0<sup>2</sup></sub> / <sub>-3<sup>3</sup></sub>	0	0	0
5	-3	-3	-3	0	0	0
6	-3	-3	-3	0 <sup>1</sup> / <sub>0<sup>2</sup></sub> / <sub>-3<sup>3</sup></sub>	0	0
7	-3	-3	-3	-3	0	0
8	-3	-3	-3	-3	0	0
9	-3 <sup>1</sup> / <sub>-6<sup>2</sup></sub> / <sub>-6<sup>3</sup></sub>	-3	-3	-3	0 <sup>1</sup> / <sub>0<sup>2</sup></sub> / <sub>-3<sup>3</sup></sub>	0
10	-3 <sup>1</sup> / <sub>-6<sup>2</sup></sub> / <sub>-6<sup>3</sup></sub>	-3	-3	-3	-3	-3
11	-6	-3	-3	-3	-3	-3
12	-6	-3	-3	-3	-3	-3
13	-6	-3 <sup>1</sup> / <sub>-3<sup>2</sup></sub> / <sub>-6<sup>3</sup></sub>	-3	-3	-3	-3
14	-6	-3 <sup>1</sup> / <sub>-3<sup>2</sup></sub> / <sub>-6<sup>3</sup></sub>	-3	-3	-3	-3
15	-6	-6	-3 <sup>1</sup> / <sub>-3<sup>2</sup></sub> / <sub>-6<sup>3</sup></sub>	-3	-3	-3
16	-6	-6	-6	-3	-3	-3
17	-6	-6	-6	-3	-3	-3
18	-6 <sup>1</sup> / <sub>-8<sup>2</sup></sub> / <sub>-8<sup>3</sup></sub>	-6	-6	-3 <sup>1</sup> / <sub>-3<sup>2</sup></sub> / <sub>-6<sup>3</sup></sub>	-3	-3
19	-8	-6	-6	-6	-3	-3
20	-8	-6	-6	-6	-3	-3

1. 0 connectors  
2. 1 connector  
3. 2 connectors

Bits	Description
31	<b>PostCur1En: post-cursor 1 deemphasis enable.</b> Read-write. Cold reset: 0. 1=Post-cursor deemphasis is enabled. 0=Post-cursor 1 deemphasis is not supported.
30	<b>PostCur2En: post-cursor 2 deemphasis enable.</b> Read-write. Cold reset: 0. 1=Post-cursor deemphasis is enabled. 0=Post-cursor 2 deemphasis is not supported.
29	<b>PreCur1En: pre-cursor 1 deemphasis enable.</b> Read-write. Cold reset: 0. 1=The data path to the transmitter is delayed by one bit time in support of pre-cursor 1 deemphasis. 0=The data path to the transmitter is not delayed by one bit time; pre-cursor 1 deemphasis is not supported. If pre-cursor 1 deemphasis is not required, this bit should be left in the low state for better performance.
28:26	Must be 000b. Read-write.

25:21	<p><b>VML: transmitter voltage margin level.</b> Read-write. Cold reset: 0. 0=Voltage margining is disabled. This field specifies a reduction in the nominal output differential voltage levels, full-swing or deemphasized, as follows:</p> <ul style="list-style-type: none"> <li>• MapPreToMargin=0: Margined diff voltage = nominal diff voltage * (1 - VML/3Eh)</li> <li>• MapPreToMargin=1: Margined diff voltage = nominal diff voltage * (1 - (VML+DP1)/3Eh)</li> </ul> <p>Voltage margining controlled by this field is intended to aid in link electrical testing and characterization. Note that the actual voltage levels are subject to quantization effects and other effects that reduce the accuracy of the above equations.</p>
20:16	<p><b>DL1: deemphasis level 1.</b> Read-write. Cold reset: 12h. The cold reset value supports -3dB deemphasis level.</p>
15:14	Reserved.
13	<p><b>MapPreToMargin: map pre-cursor to margin.</b> Revision B: Reserved. Revision C and later: Read-write. Cold reset: 0. 1=Pre-cursor deemphasis is mapped to margining. See VML.</p>
12:8	<p><b>DL2: deemphasis level 2.</b> Read-write. Cold reset: 0.</p>
7	<p><b>TxLs23ClkGateEn: LS2/LS3 clock gating enable.</b> Revision B and earlier: Reserved. Revision C and later: Read-write. Cold reset: 0. 1= internal phy clock grids are gated during LS2 or PHY OFF states to save power. BIOS should set this bit for revision (DA-C    C3    D1 and later G34    D1 and later C32    E) processors when the link is operating at Gen3 speeds. BIOS should leave this bit at the default value in all other cases.</p>
6	<p><b>MapPostCur2En: Map post-cursor 2 deemphasis enable.</b> Read-write. Cold reset: 0. 1=Post-cursor 2 deemphasis is mapped to post-cursor 1 deemphasis. See above.</p>
5	<p><b>MapPreCurEn: Map pre-cursor deemphasis enable.</b> Read-write. Cold reset: 0. 1=Pre-cursor deemphasis is mapped to post-cursor 1 deemphasis. See above.</p>
4:0	<p><b>DP1: deemphasis pre-cursor level 1.</b> Read-write. Cold reset: 0.</p>

#### **F4x1[9C, 94, 8C, 84]\_x[DF, CF] Link FIFO Read Pointer Optimization Registers**

Cold reset: 0000 0000h. See [F4x1\[98, 90, 88, 80\]](#) for register access information. The \_xCF register number specifies values for CAD[7:0], CTL0, and CLK0; the \_xDF register number specifies values for CAD[15:8], CTL1, and CLK1.

There is a synchronization FIFO between the NB clock domain and each of the link clock domains. At cold reset, the read pointer and write pointer for each of these FIFOs is positioned conservatively (30 bit-times apart), such that FIFO latency may be greater than is necessary. This register may be used to position the read pointer and write pointer of each FIFO closer to each other such that latency is reduced. Each of the fields of this register specify the number of positions to move read pointer closer to the write pointer. After writing to this register, the new values are applied to the FIFOs each time the link disconnects and reconnects, including warm resets and LDTSTOP\_L assertions. Reads from the register after a write but before the link disconnects and reconnects, returns the current value, not the pending value from the last write. Async clocking mode does not move the pointers closer than programmed, it only allows them to keep the programmed separation when the received clock is faster or slower than the transmit clock.

**Table 104: Recommended XmtRdPtr settings for G34 and C32 processors**

NB COF (GHz)	HTCLK (GHz)	F4x1[9C,94,84]_x[DF,CF][XmtRdPtr]	F4x18C_x[DF,CF][XmtRdPtr]
1.4	Any	2	2
1.6	1.2	6	6
	1.4	6	5
	1.6	6	6
	1.8	5	5
	2.0	5	4
	2.2	5	4
	2.4	5	4
	2.6	5	3
	2.8	4	3
	3.0	4	3
3.2	6	4	
1.8	1.2	6	6
	1.4	6	5
	1.6	6	5
	1.8	6	6
	2.0	5	4
	2.2	5	4
	2.4	5	4
	2.6	5	3
	2.8	5	3
	3.0	4	3
3.2	4	3	
2.0	1.2	6	6
	1.4	6	5
	1.6	6	5
	1.8	6	5
	2.0	6	5
	2.2	5	4
	2.4	5	5
	2.6	5	4
	2.8	5	3
	3.0	5	3
3.2	4	3	

**Table 104: Recommended XmtRdPtr settings for G34 and C32 processors**

NB COF (GHz)	HTCLK (GHz)	F4x1[9C,94,84]_x[DF,CF][XmtRdPtr]	F4x18C_x[DF,CF][XmtRdPtr]
2.2	1.2	6	6
	1.4	6	5
	1.6	6	5
	1.8	6	5
	2.0	6	5
	2.2	6	5
	2.4	5	5
	2.6	5	4
	2.8	5	3
	3.0	5	3
	3.2	4	3

Bits	Description
31:8	Reserved.
7:4	<p><b>XmtRdPtr: transmit FIFO read pointer.</b> Read-write. Specified in double-bit time increments.</p> <p>0h Position the read pointer 0 bit times closer to the write pointer.  1h Position the read pointer 2 bit times closer to the write pointer.  ... ..  Fh Position the read pointer 30 bit times closer to the write pointer.</p> <p>For Gen 1 links, BIOS should program this to 6h.  For Gen 3 links, BIOS should program this according to the following recommendations:</p> <ul style="list-style-type: none"> <li>• Revision B and revision C if F3x1F0[NbPstate]=000b: 5h</li> <li>• Revision C if F3x1F0[NbPstate] != 000b: 4h.</li> <li>• Revision D F(1207): 5h.</li> <li>• Revision D G34 and C32: See <a href="#">Table 104</a>.</li> </ul> <p>Values less than the recommended value are allowed; values greater than the recommended value are illegal.</p>
3:0	<p><b>RcvRdPtr: receive FIFO read pointer.</b> Read-write. Specified in double-bit time increments.</p> <p>0h Position the read pointer 0 bit times closer to the write pointer.  1h Position the read pointer 2 bit times closer to the write pointer.  ... ..  Fh Position the read pointer 30 bit times closer to the write pointer.</p> <p>BIOS should program this to Dh for Gen 1 links and Ah for Gen 3 links. Values less than the recommended value are allowed; values greater than the recommended value are illegal.</p>

### F4x1[9C, 94, 8C, 84]\_xE0 Link Phy Compensation Control Register

See [F4x1\[98, 90, 88, 80\]](#) for register access information. These register bits are updated as specified by [F0x16C\[ImmUpdate\]](#).

Bits	Description
------	-------------



31:30	<p><b>CompCyc: compensation cycle.</b> Read-write. Cold reset: 0. This specifies the number of internal clock cycles used in averaging out compensation values.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Number of clocks</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>256</td> </tr> <tr> <td>01b</td> <td>128</td> </tr> <tr> <td>10b</td> <td>64</td> </tr> <tr> <td>11b</td> <td>32</td> </tr> </tbody> </table> <p>It is recommended that these bits remain in the default state.</p>	Bits	Number of clocks	00b	256	01b	128	10b	64	11b	32
Bits	Number of clocks										
00b	256										
01b	128										
10b	64										
11b	32										
29:28	Reserved.										
27:23	<b>RttRawCal: receiver termination resistance (Rtt) raw calibration value.</b> Read-only. Cold reset: X. This field provides the raw Rtt calibration value as determined by the compensation circuit.										
22:18	<b>RonRawCal: transmitter resistance (Ron) raw calibration value.</b> Read-only. Cold reset: X. This field provides the raw Ron calibration value as determined by the compensation circuit.										
17:0	Reserved.										

### F4x1[9C, 94, 8C, 84]\_x100 Link BIST Control Register

See F4x1[98, 90, 88, 80] for register access information.

Bits	Description																								
31	<b>Width.</b> Read-only. Indicates the implemented width of the BIST engine. 0=8 bits. In 16-bit links, the same patterns are transmitted on the upper and lower sublinks. The upper bit of F0x[18C:170][LaneSel] selects which half of the link is checked in the receiver.																								
30:27	Reserved.																								
26:16	<b>ErrCnt: error count.</b> Read; write-1s-only-to-clear (writes other than all-zeroes or all-ones result in undefined behavior); controlled by hardware. Cold reset: 0. This field is incremented by hardware upon detection of each error on any lane. This count is the sum of error counts from each lane, each of which saturates at 63. See F4x1[9C, 94, 8C, 84]_x144[ErrCntCtl], F4x1[9C, 94, 8C, 84]_x148, and F4x1[9C, 94, 8C, 84]_x14C.																								
15:13	Reserved.																								
12:8	<p><b>ErrLnNum: error lane number.</b> Read; write-1s-only-to-clear (writes other than all-zeroes or all-ones result in undefined behavior); controlled by hardware. Cold reset: 0. This value is set by hardware to the lane of the sublink that failed upon detection of the first error by the BIST receiver. If multiple bits fail at the same time, the highest-numbered bit is recorded.</p> <table border="1"> <thead> <tr> <th>ErrLnNum</th> <th>Lane</th> <th>ErrLnNum</th> <th>Lane</th> </tr> </thead> <tbody> <tr> <td>0000b</td> <td>CAD0</td> <td>0101b</td> <td>CAD5</td> </tr> <tr> <td>0001b</td> <td>CAD1</td> <td>0110b</td> <td>CAD6</td> </tr> <tr> <td>0010b</td> <td>CAD2</td> <td>0111b</td> <td>CAD7</td> </tr> <tr> <td>0011b</td> <td>CAD3</td> <td>1000b</td> <td>CTL</td> </tr> <tr> <td>0100b</td> <td>CAD4</td> <td></td> <td>All other encodings reserved.</td> </tr> </tbody> </table>	ErrLnNum	Lane	ErrLnNum	Lane	0000b	CAD0	0101b	CAD5	0001b	CAD1	0110b	CAD6	0010b	CAD2	0111b	CAD7	0011b	CAD3	1000b	CTL	0100b	CAD4		All other encodings reserved.
ErrLnNum	Lane	ErrLnNum	Lane																						
0000b	CAD0	0101b	CAD5																						
0001b	CAD1	0110b	CAD6																						
0010b	CAD2	0111b	CAD7																						
0011b	CAD3	1000b	CTL																						
0100b	CAD4		All other encodings reserved.																						

7:6	<b>ErrStat: error status.</b> Read; write-1s-only-to-clear (writes other than all-zeroes or all-ones result in undefined behavior); controlled by hardware. Cold reset: 00b. This value is set by hardware to the error type upon detection of the first error by the BIST receiver. <u>Bits</u> <u>Status</u> 00b no error 01b training error 10b pattern miscompare 11b reserved
5	<b>InvRotEn: inversion rotate enable.</b> Read-write. Cold reset: 0. This bit enables rotation of [The Link BIST Southbound TX Inversion Register] F4x1[9C, 94, 8C, 84]_x110 and [The Link BIST Northbound RX Inversion Register] F4x1[9C, 94, 8C, 84]_x130 at the completion of each BIST loop.
4:2	Reserved.
1	<b>RxDis: receiver disable.</b> Read-write. Cold reset: 0. 1=Disables checking of BIST patterns in the receiver if BIST is already active. An LDTSTOP# or RESET# assertion is still required to exit BIST. If BIST has not started yet, setting this bit additionally removes any dependency on receiver link training, such that the transmitter sequences through the minimum training sets and begin sending BIST patterns at the completion of these training sets.
0	Reserved.

#### **F4x1[9C, 94, 8C, 84]\_x104 Link BIST Southbound TX Pattern Control Register**

See F4x1[98, 90, 88, 80] for register access information.

Bits	Description
31:26	Reserved.
25:21	<b>ConstCnt: constant generator count.</b> Read-write. Cold reset: 0. Selects the number of times to repeat the constant selected by ConstSel, in multiples of 24 bits. 00000b: 0 (disabled) 00001b: 24 bits ... 11111b: 24*31=744 bits
20	<b>ConstSel: constant generator select.</b> Read-write. Cold reset: 0. Selects 0 or 1 to send for the time the constant generator is active.
19:13	<b>ModCnt: modulo-N count.</b> Read-write. Cold reset: 0. Selects the number of times to repeat the Modulo-N counter (a counter with a period of N bits) pattern, 0 to 127.
12:10	<b>ModSel: modulo-N select.</b> Read-write. Cold reset: 0. Selects the pattern sent by the Modulo-N counter: <u>Bits</u> <u>Divisor – Pattern</u> 001b L/2 – 0101_0101_0101_0101_0101_0101b 010b L/4 – 0011_0011_0011_0011_0011_0011b 011b L/6 – 0001_1100_0111_0001_1100_0111b 100b L/8 – 0000_1111_0000_1111_0000_1111b 110b L/24 – 0000_0000_0000_1111_1111_1111b all others reserved

9:3	<b>PatCnt: pattern buffer count.</b> Read-write. Cold reset: 0. Selects the number of times to repeat the pattern selected by F4x1[9C, 94, 8C, 84]_x118, 0 to 127.																
2:0	<b>Order.</b> Read-write. Cold reset: 0. Selects the order in which each pattern is sent. <table border="1"> <thead> <tr> <th>Bits</th> <th>Order</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Pattern Buffer, Modulo-N Counter, Constant Generator</td> </tr> <tr> <td>001b</td> <td>Pattern Buffer, Constant Generator, Modulo-N Counter</td> </tr> <tr> <td>010b</td> <td>Modulo-N Counter, Pattern Buffer, Constant Generator</td> </tr> <tr> <td>011b</td> <td>Modulo-N Counter, Constant Generator, Pattern Buffer</td> </tr> <tr> <td>100b</td> <td>Constant Generator, Pattern Buffer, Modulo-N Counter</td> </tr> <tr> <td>101b</td> <td>Constant Generator, Modulo-N Counter, Pattern Buffer</td> </tr> <tr> <td>110, 111b</td> <td>reserved</td> </tr> </tbody> </table>	Bits	Order	000b	Pattern Buffer, Modulo-N Counter, Constant Generator	001b	Pattern Buffer, Constant Generator, Modulo-N Counter	010b	Modulo-N Counter, Pattern Buffer, Constant Generator	011b	Modulo-N Counter, Constant Generator, Pattern Buffer	100b	Constant Generator, Pattern Buffer, Modulo-N Counter	101b	Constant Generator, Modulo-N Counter, Pattern Buffer	110, 111b	reserved
Bits	Order																
000b	Pattern Buffer, Modulo-N Counter, Constant Generator																
001b	Pattern Buffer, Constant Generator, Modulo-N Counter																
010b	Modulo-N Counter, Pattern Buffer, Constant Generator																
011b	Modulo-N Counter, Constant Generator, Pattern Buffer																
100b	Constant Generator, Pattern Buffer, Modulo-N Counter																
101b	Constant Generator, Modulo-N Counter, Pattern Buffer																
110, 111b	reserved																

### F4x1[9C, 94, 8C, 84]\_x108 Link BIST Southbound TX Pattern Buffer 1 Register

See F4x1[98, 90, 88, 80] for register access information.

Bits	Description
31:24	Reserved.
23:0	<b>Pattern1[23:0].</b> Read-write. Cold reset: 0. Holds the first 24 bits of Pattern Buffer 1.

### F4x1[9C, 94, 8C, 84]\_x10C Link BIST Southbound TX Mask Register

See F4x1[98, 90, 88, 80] for register access information.

Bits	Description																								
31:9	Reserved.																								
8:0	<b>TxMask[8:0].</b> Read-write. Cold reset: 1FFh. Selects lanes of the sublinks to transmit a logical 0. 1=Lane active. 0=Lane masked. <table border="1"> <thead> <tr> <th>Bit</th> <th>Lane</th> <th>Bit</th> <th>Lane</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>CAD0</td> <td>5</td> <td>CAD5</td> </tr> <tr> <td>1</td> <td>CAD1</td> <td>6</td> <td>CAD6</td> </tr> <tr> <td>2</td> <td>CAD2</td> <td>7</td> <td>CAD7</td> </tr> <tr> <td>3</td> <td>CAD3</td> <td>8</td> <td>CTL</td> </tr> <tr> <td>4</td> <td>CAD4</td> <td></td> <td></td> </tr> </tbody> </table>	Bit	Lane	Bit	Lane	0	CAD0	5	CAD5	1	CAD1	6	CAD6	2	CAD2	7	CAD7	3	CAD3	8	CTL	4	CAD4		
Bit	Lane	Bit	Lane																						
0	CAD0	5	CAD5																						
1	CAD1	6	CAD6																						
2	CAD2	7	CAD7																						
3	CAD3	8	CTL																						
4	CAD4																								

**F4x1[9C, 94, 8C, 84]\_x110 Link BIST Southbound TX Inversion Register**

See F4x1[98, 90, 88, 80] for register access information.

Bits	Description																								
31:9	Reserved.																								
8:0	<p><b>TxInv[8:0]</b>. Read-write. Cold reset: 0. Selects lanes of the sublinks to invert. 1=Lane inverted. 0=Lane unmodified.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Lane</th> <th>Bit</th> <th>Lane</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>CAD0</td> <td>5</td> <td>CAD5</td> </tr> <tr> <td>1</td> <td>CAD1</td> <td>6</td> <td>CAD6</td> </tr> <tr> <td>2</td> <td>CAD2</td> <td>7</td> <td>CAD7</td> </tr> <tr> <td>3</td> <td>CAD3</td> <td>8</td> <td>CTL</td> </tr> <tr> <td>4</td> <td>CAD4</td> <td></td> <td></td> </tr> </tbody> </table> <p>When F4x1[9C, 94, 8C, 84]_x100[InvRotEn] is set, the bits corresponding to active lanes rotate to the left at the completion of each BIST loop: {NxtTxInv[8:0]}={TxInv[7:0],TxInv[8]}. Note: if the transmitter and receiver are different widths, inversion rotation can only be used for 16/8-bit links and the initial pattern in the inversion register must repeat on 9-bit boundaries.</p>	Bit	Lane	Bit	Lane	0	CAD0	5	CAD5	1	CAD1	6	CAD6	2	CAD2	7	CAD7	3	CAD3	8	CTL	4	CAD4		
Bit	Lane	Bit	Lane																						
0	CAD0	5	CAD5																						
1	CAD1	6	CAD6																						
2	CAD2	7	CAD7																						
3	CAD3	8	CTL																						
4	CAD4																								

**F4x1[9C, 94, 8C, 84]\_x114 Link BIST Southbound TX Pattern Buffer 2 Register**

See F4x1[98, 90, 88, 80] for register access information.

Bits	Description
31:24	Reserved.
23:0	<b>Pattern2[23:0]</b> . Read-write. Cold reset: 0. Holds the first 24 bits of Pattern Buffer 2.

**F4x1[9C, 94, 8C, 84]\_x118 Link BIST Southbound TX Pattern Buffer 2 Enable Register**

See F4x1[98, 90, 88, 80] for register access information.

Bits	Description																								
31:9	Reserved.																								
8:0	<p><b>Pat2En[8:0]</b>. Read-write. Cold reset: 0. Selects lanes of the sublinks that use Pattern Buffer 2 instead of Pattern Buffer 1. 1=Buffer 2 selected. 0=Buffer 1 selected.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Lane</th> <th>Bit</th> <th>Lane</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>CAD0</td> <td>5</td> <td>CAD5</td> </tr> <tr> <td>1</td> <td>CAD1</td> <td>6</td> <td>CAD6</td> </tr> <tr> <td>2</td> <td>CAD2</td> <td>7</td> <td>CAD7</td> </tr> <tr> <td>3</td> <td>CAD3</td> <td>8</td> <td>CTL</td> </tr> <tr> <td>4</td> <td>CAD4</td> <td></td> <td></td> </tr> </tbody> </table>	Bit	Lane	Bit	Lane	0	CAD0	5	CAD5	1	CAD1	6	CAD6	2	CAD2	7	CAD7	3	CAD3	8	CTL	4	CAD4		
Bit	Lane	Bit	Lane																						
0	CAD0	5	CAD5																						
1	CAD1	6	CAD6																						
2	CAD2	7	CAD7																						
3	CAD3	8	CTL																						
4	CAD4																								

**F4x1[9C, 94, 8C, 84]\_x11C Link BIST Southbound TX Pattern Buffer Extension Register**

See F4x1[98, 90, 88, 80] for register access information.

Bits	Description
31:16	<b>Pattern2[39:24]</b> . Read-write. Cold reset: 0. Holds the upper 16 bits of Pattern Buffer 2.
15:0	<b>Pattern1[39:24]</b> . Read-write. Cold reset: 0. Holds the upper 16 bits of Pattern Buffer 1.

**F4x1[9C, 94, 8C, 84]\_x120 Link BIST Southbound TX Scramble Register**

See F4x1[98, 90, 88, 80] for register access information.

Bits	Description																								
31:9	Reserved.																								
8:0	<p><b>TxScramble.</b> Read-write. Cold reset: 0. Selects lanes of the sublinks to scramble. 1=Scrambling enabled. 0=Scrambling disabled.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Lane</th> <th>Bit</th> <th>Lane</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>CAD0</td> <td>5</td> <td>CAD5</td> </tr> <tr> <td>1</td> <td>CAD1</td> <td>6</td> <td>CAD6</td> </tr> <tr> <td>2</td> <td>CAD2</td> <td>7</td> <td>CAD7</td> </tr> <tr> <td>3</td> <td>CAD3</td> <td>8</td> <td>CTL</td> </tr> <tr> <td>4</td> <td>CAD4</td> <td></td> <td></td> </tr> </tbody> </table>	Bit	Lane	Bit	Lane	0	CAD0	5	CAD5	1	CAD1	6	CAD6	2	CAD2	7	CAD7	3	CAD3	8	CTL	4	CAD4		
Bit	Lane	Bit	Lane																						
0	CAD0	5	CAD5																						
1	CAD1	6	CAD6																						
2	CAD2	7	CAD7																						
3	CAD3	8	CTL																						
4	CAD4																								

**F4x1[9C, 94, 8C, 84]\_x124 Link BIST Northbound RX Pattern Control Register**

See F4x1[98, 90, 88, 80] for register access information.

Bits	Description														
31:26	Reserved.														
25:21	<p><b>ConstCnt: constant generator count.</b> Read-write. Cold reset: 0. Selects the number of times to repeat the constant selected by ConstSel, in multiples of 24 bits.</p> <p>00000b: 0 (disabled)  00001b: 24 bits  ...  11111b: 24*31=744 bits</p>														
20	<p><b>ConstSel: constant generator select.</b> Read-write. Cold reset: 0. Selects 0 or 1 to send for the time the constant generator is active.</p>														
19:13	<p><b>ModCnt: modulo-N count.</b> Read-write. Cold reset: 0. Selects the number of times to repeat the Modulo-N counter (a counter with a period of N bits) pattern, 0 to 127.</p>														
12:10	<p><b>ModSel: modulo-N select.</b> Read-write. Cold reset: 0. Selects the pattern sent by the Modulo-N counter:</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Divisor – Pattern</th> </tr> </thead> <tbody> <tr> <td>001b</td> <td>L/2 – 0101_0101_0101_0101_0101_0101b</td> </tr> <tr> <td>010b</td> <td>L/4 – 0011_0011_0011_0011_0011_0011b</td> </tr> <tr> <td>011b</td> <td>L/6 – 0001_1100_0111_0001_1100_0111b</td> </tr> <tr> <td>100b</td> <td>L/8 – 0000_1111_0000_1111_0000_1111b</td> </tr> <tr> <td>110b</td> <td>L/24 – 0000_0000_0000_1111_1111_1111b</td> </tr> <tr> <td>all others</td> <td>reserved</td> </tr> </tbody> </table>	Bits	Divisor – Pattern	001b	L/2 – 0101_0101_0101_0101_0101_0101b	010b	L/4 – 0011_0011_0011_0011_0011_0011b	011b	L/6 – 0001_1100_0111_0001_1100_0111b	100b	L/8 – 0000_1111_0000_1111_0000_1111b	110b	L/24 – 0000_0000_0000_1111_1111_1111b	all others	reserved
Bits	Divisor – Pattern														
001b	L/2 – 0101_0101_0101_0101_0101_0101b														
010b	L/4 – 0011_0011_0011_0011_0011_0011b														
011b	L/6 – 0001_1100_0111_0001_1100_0111b														
100b	L/8 – 0000_1111_0000_1111_0000_1111b														
110b	L/24 – 0000_0000_0000_1111_1111_1111b														
all others	reserved														

9:3	<b>PatCnt: pattern buffer count.</b> Read-write. Cold reset: 0. Selects the number of times to repeat the pattern selected by <code>F4x1[9C, 94, 8C, 84]_x118</code> , 0 to 127.																
2:0	<b>Order.</b> Read-write. Cold reset: 0. Selects the order in which each pattern is sent. <table border="1"> <thead> <tr> <th>Bits</th> <th>Order</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Pattern Buffer, Modulo-N Counter, Constant Generator</td> </tr> <tr> <td>001b</td> <td>Pattern Buffer, Constant Generator, Modulo-N Counter</td> </tr> <tr> <td>010b</td> <td>Modulo-N Counter, Pattern Buffer, Constant Generator</td> </tr> <tr> <td>011b</td> <td>Modulo-N Counter, Constant Generator, Pattern Buffer</td> </tr> <tr> <td>100b</td> <td>Constant Generator, Pattern Buffer, Modulo-N Counter</td> </tr> <tr> <td>101b</td> <td>Constant Generator, Modulo-N Counter, Pattern Buffer</td> </tr> <tr> <td>110, 111b</td> <td>reserved</td> </tr> </tbody> </table>	Bits	Order	000b	Pattern Buffer, Modulo-N Counter, Constant Generator	001b	Pattern Buffer, Constant Generator, Modulo-N Counter	010b	Modulo-N Counter, Pattern Buffer, Constant Generator	011b	Modulo-N Counter, Constant Generator, Pattern Buffer	100b	Constant Generator, Pattern Buffer, Modulo-N Counter	101b	Constant Generator, Modulo-N Counter, Pattern Buffer	110, 111b	reserved
Bits	Order																
000b	Pattern Buffer, Modulo-N Counter, Constant Generator																
001b	Pattern Buffer, Constant Generator, Modulo-N Counter																
010b	Modulo-N Counter, Pattern Buffer, Constant Generator																
011b	Modulo-N Counter, Constant Generator, Pattern Buffer																
100b	Constant Generator, Pattern Buffer, Modulo-N Counter																
101b	Constant Generator, Modulo-N Counter, Pattern Buffer																
110, 111b	reserved																

### **F4x1[9C, 94, 8C, 84]\_x128 Link BIST Northbound RX Pattern Buffer 1 Register**

See `F4x1[98, 90, 88, 80]` for register access information.

Bits	Description
31:24	Reserved.
23:0	<b>Pattern1[23:0].</b> Read-write. Cold reset: 0. Holds the first 24 bits of Pattern Buffer 1.

### **F4x1[9C, 94, 8C, 84]\_x12C Link BIST Northbound RX Mask Register**

See `F4x1[98, 90, 88, 80]` for register access information.

Bits	Description																								
31:9	Reserved.																								
8:0	<b>RxMask[8:0].</b> Read-write. Cold reset: 1FFh. Selects lanes of the selected sublink that are checked by the receiver. 1=Lane active. 0=Lane masked. Software is responsible for clearing bits 7:4 for 4-bit links and bits 7:2 for 2-bit links. <table border="1"> <thead> <tr> <th>Bit</th> <th>Lane</th> <th>Bit</th> <th>Lane</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>CAD0</td> <td>5</td> <td>CAD5</td> </tr> <tr> <td>1</td> <td>CAD1</td> <td>6</td> <td>CAD6</td> </tr> <tr> <td>2</td> <td>CAD2</td> <td>7</td> <td>CAD7</td> </tr> <tr> <td>3</td> <td>CAD3</td> <td>8</td> <td>CTL</td> </tr> <tr> <td>4</td> <td>CAD4</td> <td></td> <td></td> </tr> </tbody> </table>	Bit	Lane	Bit	Lane	0	CAD0	5	CAD5	1	CAD1	6	CAD6	2	CAD2	7	CAD7	3	CAD3	8	CTL	4	CAD4		
Bit	Lane	Bit	Lane																						
0	CAD0	5	CAD5																						
1	CAD1	6	CAD6																						
2	CAD2	7	CAD7																						
3	CAD3	8	CTL																						
4	CAD4																								

**F4x1[9C, 94, 8C, 84]\_x130 Link BIST Northbound RX Inversion Register**

See F4x1[98, 90, 88, 80] for register access information.

Bits	Description																								
31:9	Reserved.																								
8:0	<p><b>RxInv[8:0]</b>. Read-write. Cold reset: 0. Selects lanes of the sublink that are inverted. 1=Lane inverted. 0=Lane unmodified.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Lane</th> <th>Bit</th> <th>Lane</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>CAD0</td> <td>5</td> <td>CAD5</td> </tr> <tr> <td>1</td> <td>CAD1</td> <td>6</td> <td>CAD6</td> </tr> <tr> <td>2</td> <td>CAD2</td> <td>7</td> <td>CAD7</td> </tr> <tr> <td>3</td> <td>CAD3</td> <td>8</td> <td>CTL</td> </tr> <tr> <td>4</td> <td>CAD4</td> <td></td> <td></td> </tr> </tbody> </table> <p>When F4x1[9C, 94, 8C, 84]_x100[InvRotEn] is set, the bits corresponding to active lanes rotate to the left at the completion of each BIST loop: {NxtTxInv[8:0]}={TxInv[7:0],TxInv[8]}. Note: if the transmitter and receiver are different widths, inversion rotation can only be used for 16/8-bit links and the initial pattern in the inversion register must repeat on 9-bit boundaries.</p>	Bit	Lane	Bit	Lane	0	CAD0	5	CAD5	1	CAD1	6	CAD6	2	CAD2	7	CAD7	3	CAD3	8	CTL	4	CAD4		
Bit	Lane	Bit	Lane																						
0	CAD0	5	CAD5																						
1	CAD1	6	CAD6																						
2	CAD2	7	CAD7																						
3	CAD3	8	CTL																						
4	CAD4																								

**F4x1[9C, 94, 8C, 84]\_x134 Link BIST Northbound RX Pattern Buffer 2 Register**

See F4x1[98, 90, 88, 80] for register access information.

Bits	Description
31:24	Reserved.
23:0	<b>Pattern2[23:0]</b> . Read-write. Cold reset: 0. Holds the first 24 bits of Pattern Buffer 2.

**F4x1[9C, 94, 8C, 84]\_x138 Link BIST Northbound RX Pattern Buffer 2 Enable Register**

See F4x1[98, 90, 88, 80] for register access information.

Bits	Description																								
31:9	Reserved.																								
8:0	<p><b>Pat2En[8:0]</b>. Read-write. Cold reset: 0. Selects lanes of the sublink that use Pattern Buffer 2 instead of Pattern Buffer 1. 1=Buffer 2 selected. 0=Buffer 1 selected.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Lane</th> <th>Bit</th> <th>Lane</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>CAD0</td> <td>5</td> <td>CAD5</td> </tr> <tr> <td>1</td> <td>CAD1</td> <td>6</td> <td>CAD6</td> </tr> <tr> <td>2</td> <td>CAD2</td> <td>7</td> <td>CAD7</td> </tr> <tr> <td>3</td> <td>CAD3</td> <td>8</td> <td>CTL</td> </tr> <tr> <td>4</td> <td>CAD4</td> <td></td> <td></td> </tr> </tbody> </table>	Bit	Lane	Bit	Lane	0	CAD0	5	CAD5	1	CAD1	6	CAD6	2	CAD2	7	CAD7	3	CAD3	8	CTL	4	CAD4		
Bit	Lane	Bit	Lane																						
0	CAD0	5	CAD5																						
1	CAD1	6	CAD6																						
2	CAD2	7	CAD7																						
3	CAD3	8	CTL																						
4	CAD4																								

**F4x1[9C, 94, 8C, 84]\_x13C Link BIST Northbound RX Pattern Buffer Extension Register**

See F4x1[98, 90, 88, 80] for register access information.

Bits	Description
31:16	<b>Pattern2[39:24]</b> . Read-write. Cold reset: 0. Holds the upper 16 bits of Pattern Buffer 2.
15:0	<b>Pattern1[39:24]</b> . Read-write. Cold reset: 0. Holds the upper 16 bits of Pattern Buffer 1.



**F4x1[9C, 94, 8C, 84]\_x140 Link BIST Northbound RX Scramble Register**

See F4x1[98, 90, 88, 80] for register access information.

Bits	Description																								
31:9	Reserved.																								
8:0	<p><b>RxScramble.</b> Read-write. Cold reset: 0. Selects lanes of the sublink to scramble. 1=Scrambling enabled. 0=Scrambling disabled.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Lane</th> <th>Bit</th> <th>Lane</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>CAD0</td> <td>5</td> <td>CAD5</td> </tr> <tr> <td>1</td> <td>CAD1</td> <td>6</td> <td>CAD6</td> </tr> <tr> <td>2</td> <td>CAD2</td> <td>7</td> <td>CAD7</td> </tr> <tr> <td>3</td> <td>CAD3</td> <td>8</td> <td>CTL</td> </tr> <tr> <td>4</td> <td>CAD4</td> <td></td> <td></td> </tr> </tbody> </table>	Bit	Lane	Bit	Lane	0	CAD0	5	CAD5	1	CAD1	6	CAD6	2	CAD2	7	CAD7	3	CAD3	8	CTL	4	CAD4		
Bit	Lane	Bit	Lane																						
0	CAD0	5	CAD5																						
1	CAD1	6	CAD6																						
2	CAD2	7	CAD7																						
3	CAD3	8	CTL																						
4	CAD4																								

**F4x1[9C, 94, 8C, 84]\_x144 Link BIST Northbound RX Error Status Register**

See F4x1[98, 90, 88, 80] for register access information.

Bits	Description																								
31	Reserved.																								
30:24	<p><b>ErrCntCtl: CTL lane Error Count.</b> Revision C and earlier revisions: Reserved. Revision D and later: Read-only. ErrCntCtl[5:0] indicates the number of BIST errors detected on this lane since the last time F4x1[9C, 94, 8C, 84]_x100[ErrCnt] was cleared. ErrCntCtl[6] indicates overflow when set.</p>																								
23:9	Reserved.																								
8:0	<p><b>RxErrStat.</b> Read; write-0-to-clear (all bits of the field must be 0; if any of them are set, the write is ignored); set-by-hardware. Cold reset: 0. Indicates lanes of the selected sublink that had errors.</p> <table border="1"> <thead> <tr> <th>Bit</th> <th>Lane</th> <th>Bit</th> <th>Lane</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>CAD0</td> <td>5</td> <td>CAD5</td> </tr> <tr> <td>1</td> <td>CAD1</td> <td>6</td> <td>CAD6</td> </tr> <tr> <td>2</td> <td>CAD2</td> <td>7</td> <td>CAD7</td> </tr> <tr> <td>3</td> <td>CAD3</td> <td>8</td> <td>CTL</td> </tr> <tr> <td>4</td> <td>CAD4</td> <td></td> <td></td> </tr> </tbody> </table>	Bit	Lane	Bit	Lane	0	CAD0	5	CAD5	1	CAD1	6	CAD6	2	CAD2	7	CAD7	3	CAD3	8	CTL	4	CAD4		
Bit	Lane	Bit	Lane																						
0	CAD0	5	CAD5																						
1	CAD1	6	CAD6																						
2	CAD2	7	CAD7																						
3	CAD3	8	CTL																						
4	CAD4																								

**F4x1[9C, 94, 8C, 84]\_x148 Link BIST Northbound RX Per-Lane Error Count Register 1**

Revision C and earlier revisions: Reserved. Revision D and later: See F4x1[98, 90, 88, 80] for register access information.

Bits	Description
31	Reserved.
30:24	<b>ErrCntCad3: CAD3 lane Error Count.</b> See F4x1[9C, 94, 8C, 84]_x144[ErrCntCtl].
23	Reserved.
22:16	<b>ErrCntCad2: CAD2 lane Error Count.</b> See F4x1[9C, 94, 8C, 84]_x144[ErrCntCtl].
15	Reserved.
14:8	<b>ErrCntCad1: CAD1 lane Error Count.</b> See F4x1[9C, 94, 8C, 84]_x144[ErrCntCtl].

7	Reserved.
6:0	<b>ErrCntCad0: CAD0 lane Error Count.</b> See F4x1[9C, 94, 8C, 84]_x144[ErrCntCtl].

### **F4x1[9C, 94, 8C, 84]\_x14C Link BIST Northbound RX Per-Lane Error Count Register 2**

Revision C and earlier revisions: Reserved. Revision D and later: See F4x1[98, 90, 88, 80] for register access information.

Bits	Description
31	Reserved.
30:24	<b>ErrCntCad7: CAD7 lane Error Count.</b> See F4x1[9C, 94, 8C, 84]_x144[ErrCntCtl].
23	Reserved.
22:16	<b>ErrCntCad6: CAD6 lane Error Count.</b> See F4x1[9C, 94, 8C, 84]_x144[ErrCntCtl].
15	Reserved.
14:8	<b>ErrCntCad5: CAD5 lane Error Count.</b> See F4x1[9C, 94, 8C, 84]_x144[ErrCntCtl].
7	Reserved.
6:0	<b>ErrCntCad4: CAD4 lane Error Count.</b> See F4x1[9C, 94, 8C, 84]_x144[ErrCntCtl].

### **F4x1[9C, 94, 8C, 84]\_x4[9:0][8, 0]6 Link Phy DFE and DFR Control Registers**

Revision C and earlier revisions: Reserved. Revision D and later revisions: The definition of this register is similar to that of F4x1[9C, 94, 8C, 84]\_x[D4, C4]. This register provides per-lane programmability whereas F4x1[9C, 94, 8C, 84]\_x[D4, C4] only allows per-sublink programmability.

This is a direct map register set. See F4x1[98, 90, 88, 80] for direct map register access information. The 16 bit wide register numbers are mapped to the transmit lanes as follows:

<u>Offset</u>	<u>Lane</u>
4006h	CADIN[0]
4086h	CADIN[1]
4106h	CADIN[2]
4186h	CADIN[3]
4206h	CADIN[4]
4286h	CADIN[5]
4306h	CADIN[6]
4386h	CADIN[7]
4406h	CADIN[8]
4486h	CADIN[9]
4506h	CADIN[10]
4586h	CADIN[11]
4606h	CADIN[12]
4686h	CADIN[13]
4706h	CADIN[14]
4786h	CADIN[15]
4806h	CTLIN[0]
4906h	CTLIN[1]

Accessing any register number that is not listed above might result in undefined behavior of the phy. These register bits are updated as specified by F0x16C[ImmUpdate].

Bits	Description
31:0	See F4x1[9C, 94, 8C, 84]_x[D4, C4].

### F4x1[9C, 94, 8C, 84]\_x[530A, 520A] DLL Control and Test Register 3

These registers are direct mapped registers, see F4x1[98, 90, 88, 80] for direct map register access information. The \_x520A register number specifies values for CAD[7:0], and CTL0; the \_x530A register number specifies values for CAD[15:8], and CTL1.

Bits	Description																
31:29	<p><b>Ls2ExitTime: LS2 exit time.</b> Read-write. Cold reset: 0. This field selects the internal timer that delays the turn-on of the DLL after exit from LS2 state to L0 state. The added delay allows the forwarded input clock to achieve better stability.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Delay=10us.</td> </tr> <tr> <td>001b</td> <td>Delay=5us.</td> </tr> <tr> <td>010b</td> <td>Delay=2.5us.</td> </tr> <tr> <td>011b</td> <td>Delay=1.25us.</td> </tr> <tr> <td>100b</td> <td>Delay=625ns.</td> </tr> <tr> <td>101b</td> <td>Delay=0s.</td> </tr> <tr> <td>110b,111b</td> <td>reserved.</td> </tr> </tbody> </table> <p>Note: The value specified by Ls2ExitTime must be less than the value specified by F0x16C[T0Time], or it can cause undefined behavior.</p>	Bits	Definition	000b	Delay=10us.	001b	Delay=5us.	010b	Delay=2.5us.	011b	Delay=1.25us.	100b	Delay=625ns.	101b	Delay=0s.	110b,111b	reserved.
Bits	Definition																
000b	Delay=10us.																
001b	Delay=5us.																
010b	Delay=2.5us.																
011b	Delay=1.25us.																
100b	Delay=625ns.																
101b	Delay=0s.																
110b,111b	reserved.																
28:15	Reserved.																
14:13	<p><b>AnalogWaitTime: analog wait time to turn on DLL.</b> Read-write. Cold reset: 0. Revisions A and B: This field is used with DllAnalogOkIgnore; if DllAnalogOkIgnore is set, the turning on of the DLL circuit after cold reset is delayed by a timer specified by this field. Revisions C and later: The turning on of the DLL circuit after cold reset is always delayed by a timer specified by this field. The encodings are as follows:</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Definition</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>Delay=1.25us.</td> </tr> <tr> <td>01b</td> <td>Delay=0.625us.</td> </tr> <tr> <td>10b</td> <td>Delay=2.5us.</td> </tr> <tr> <td>11b</td> <td>Delay=0.3125us.</td> </tr> </tbody> </table> <p>BIOS should program this field to 10b.</p>	Bits	Definition	00b	Delay=1.25us.	01b	Delay=0.625us.	10b	Delay=2.5us.	11b	Delay=0.3125us.						
Bits	Definition																
00b	Delay=1.25us.																
01b	Delay=0.625us.																
10b	Delay=2.5us.																
11b	Delay=0.3125us.																
12:11	Reserved.																
10	<p><b>DllAnalogOkIgnore: DLL analog start signal ignore.</b> Revision B: Read-write. Cold reset: 0. 1=The delay of turning on of DLL circuit after reset is controlled purely by a timer specified by AnalogWaitTime. See AnalogWaitTime for more information. 0=DLL is turned on after reset by a signal automatically generated based on the status of internal supply voltage level. BIOS should set this bit to 1b. Revision C and later: Reserved.</p>																
9:8	Reserved.																
7	<p><b>BiasDisInLs2: bias disable in LS2 power state.</b> Read-write. Cold reset: 0. 1=Enables lower power LS2 state; current consumption is lowered by approximately 2.5mA per receive lane when compared to standard LS2 power mode. Setting this bit increases the amount of T0Time needed to relock the DLL.Note: When this bit is set, Ls2ExitTime must be programmed to select a value that is greater than or equal to AnalogWaitTime. 0=Standard LS2 power mode. .</p>																

6:5	Reserved.
4	<b>LockDetOnLs2Exit: DLL lock detect on LS2 exit.</b> Read-write. Cold reset: 0. This field selects the LS2 to L0 power state transition speed. 1=Fast transition mode selected. 0=Slow transition mode selected.
3:1	Reserved.
0	<b>EnCoreLoopFirst: enable DLL core loop first on LS2 exit.</b> Read-write. Cold reset: 0. This field selects LS2 to L0 power state transition speed. 1=Fast transition mode selected. 0=Slow transition mode selected.

### **F4x1[9C, 94, 8C, 84]\_x6[9:0][8, 0]0 Link Phy Deemphasis Value Registers**

Revision C and earlier revisions: Reserved. Revision D and later: The definition of this register is similar to that of [F4x1\[9C, 94, 8C, 84\]\\_x\[D5, C5\]](#). This register provides per-lane programmability whereas [F4x1\[9C, 94, 8C, 84\]\\_x\[D5, C5\]](#) only allows per-sublink programmability. This is a direct map register set. See [F4x1\[98, 90, 88, 80\]](#) for direct map register access information. The 16 bit wide register numbers are mapped to the transmit lanes as follows:

<u>Offset</u>	<u>Lane</u>
6000h	CADOUT[0]
6080h	CADOUT[1]
6100h	CADOUT[2]
6180h	CADOUT[3]
6200h	CADOUT[4]
6280h	CADOUT[5]
6300h	CADOUT[6]
6380h	CADOUT[7]
6400h	CADOUT[8]
6480h	CADOUT[9]
6500h	CADOUT[10]
6580h	CADOUT[11]
6600h	CADOUT[12]
6680h	CADOUT[13]
6700h	CADOUT[14]
6780h	CADOUT[15]
6800h	CTL0UT[0]
6880h	CLKOUT[0]
6900h	CTL0UT[1]
6980h	CLKOUT[1]

Accessing any register number that is not listed above might result in undefined behavior of the phy. These register bits are updated as specified by [F0x16C\[ImmUpdate\]](#).

Bits	Description
31:0	See <a href="#">F4x1[9C, 94, 8C, 84]_x[D5, C5]</a> .

### **F4x1[9C, 94, 8C, 84]\_x6[9, 8]84 Link Phy Transmit Clock Phase Control Registers**

Revision B and earlier revisions: Reserved. Revision C and later revisions: This is a direct map register set, See [F4x1\[98, 90, 88, 80\]](#) for direct map register access information. The 16 bit wide register numbers are mapped

to the transmit clock lanes as follows:

<u>Offset</u>	<u>Lane</u>
6884h	CLKOUT[0]
6984h	CLKOUT[1]

Accessing any register number that is not listed above might result in undefined behavior of the phy. These register bits are updated as specified by F0x16C[ImmUpdate].

Note: This register is not applicable for CAD/CTL lanes.

Bits	Description
31:1	Reserved.
0	<b>Tx90En: transmit clock 90 degree phase shift enable.</b> Read-write. Cold reset: 1. 1=Enable 90 degree phase shift on transmit clock lanes relative to CAD/CTL lanes. 0=Clock and CAD/CTL lanes are transmitted in phase. This feature may reduce cross-talk.

#### **F4x1C4 L3 Power Control Register**

Revision B and earlier revisions: Reserved. Revision C and later revisions: Read-write. Reset: 0000 0010h.

Bits	Description														
31:9	Reserved.														
8	<b>L3PwrSavEn: L3 power savings enable.</b> 1=The clock to idle subcaches in the L3 is stopped. BIOS should set this bit.														
7:0	<b>L3ClkHysCtl: L3 clock hysteresis control.</b> Specifies the maximum number of clocks that the clock is allowed to stop for an idle subcache. After the hysteresis counter expires the clock for the idle subcache is restarted for one clock. If the subcache is still idle after sending one clock to the subcache, the clock is stopped and the hysteresis counter is restarted. <table> <thead> <tr> <th><u>Value</u></th> <th><u>Clocks</u></th> </tr> </thead> <tbody> <tr> <td>00h</td> <td>Reserved</td> </tr> <tr> <td>01h</td> <td>1</td> </tr> <tr> <td>02h</td> <td>2</td> </tr> <tr> <td>...</td> <td>...</td> </tr> <tr> <td>10h</td> <td>16</td> </tr> <tr> <td>FFh-11h</td> <td>Reserved</td> </tr> </tbody> </table> BIOS should not change this field from the default state.	<u>Value</u>	<u>Clocks</u>	00h	Reserved	01h	1	02h	2	...	...	10h	16	FFh-11h	Reserved
<u>Value</u>	<u>Clocks</u>														
00h	Reserved														
01h	1														
02h	2														
...	...														
10h	16														
FFh-11h	Reserved														

#### **F4x1C8 L3 Hit Statistics Register**

Revision C and earlier revisions: Reserved. Revision D and later: Read-write. Reset: 0000 0000h.

Bits	Description
31:28	Reserved.
27:4	<b>L3HitStat: L3 hit statistics.</b> Read-only. See StatSel for the definition of the statistics returned by this field.

3	<b>StatSel: L3 hit statistics select.</b> Read-write. 1=L3HitStat[23:0] returns a moving average of the number of L3 hits per $6 * 2^{26}$ Northbridge clocks. 0=L3HitStat[9:0] returns a moving average of the number of L3 hits per $2^{10}$ L3 accesses.														
2:0	<b>CoreSel: core select.</b> Read-write. This field specifies the core selected for L3 hit statistics. <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Core 0.</td> </tr> <tr> <td>001b</td> <td>Core 1.</td> </tr> <tr> <td>010b</td> <td>Core 2.</td> </tr> <tr> <td>011b</td> <td>Core 3.</td> </tr> <tr> <td>100b</td> <td>Core 4.</td> </tr> <tr> <td>101b</td> <td>Core 5.</td> </tr> </tbody> </table> All other encodings are reserved.	Bits	Description	000b	Core 0.	001b	Core 1.	010b	Core 2.	011b	Core 3.	100b	Core 4.	101b	Core 5.
Bits	Description														
000b	Core 0.														
001b	Core 1.														
010b	Core 2.														
011b	Core 3.														
100b	Core 4.														
101b	Core 5.														

### F4x1[F0:E0] P-state Specification Registers

All fields are read-only. These registers specify the reset defaults for fields in [The P-State [4:0] Registers] MSRC001\_00[68:64]. F4x1E0 corresponds to MSRC001\_0064; F4x1E4 corresponds to MSRC001\_0065; etc.

Bits	Description
31:28	Reserved.
27	<b>PstateEn.</b> Default for [The P-State [4:0] Registers] MSRC001_00[68:64][PstateEn].
26:25	<b>IddDiv.</b> Default for [The P-State [4:0] Registers] MSRC001_00[68:64][IddDiv].
24:17	<b>IddValue.</b> Default for [The P-State [4:0] Registers] MSRC001_00[68:64][IddValue].
16	<b>NbDid.</b> Default for [The P-State [4:0] Registers] MSRC001_00[68:64][NbDid].
15:9	<b>CpuVid.</b> Default for [The P-State [4:0] Registers] MSRC001_00[68:64][CpuVid].
8:6	<b>CpuDid.</b> Default for [The P-State [4:0] Registers] MSRC001_00[68:64][CpuDid].
5:0	<b>CpuFid.</b> Default for [The P-State [4:0] Registers] MSRC001_00[68:64][CpuFid].

### F4x1F4 Northbridge P-state Specification Register

All fields are read-only.

Bits	Description
31:14	Reserved.
13:7	<b>NbVid1.</b> This specifies the cold-reset default for the [The P-State [4:0] Registers] MSRC001_00[68:64][NbVid] fields in which MSRC001_00[68:64][NbDid] =1.
6:0	<b>NbVid0.</b> This specifies the cold-reset default for the [The P-State [4:0] Registers] MSRC001_00[68:64][NbVid] fields in which MSRC001_00[68:64][NbDid] =0.

## 3.8 APIC Registers

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention.

**APIC20 APIC ID Register**

Reset: ??00 0000h.

Bits	Description
31:24	<b>ApicId</b> . Read-write. Reset: varies based on core number and node number; see <a href="#">MSRC001_001F</a> [InitApicIdCpuIdLo]. See <a href="#">2.9.5.1 [ApicId Enumeration Requirements]</a> . When <a href="#">F0x68</a> [ApicExtId and ApicExtBrdCst] = 11b, all 8 bits of this field are used; if either of these bits is low, then bits[3:0] of this field are used and bits[7:4] are reserved. See <a href="#">2.9.2 [CPU Cores and Downcoring]</a> .
23:0	Reserved.

**APIC30 APIC Version Register**

Reset: 80?? 0010h.

Bits	Description
31	<b>ExtApicSpace: extended APIC register space present</b> . Read-only. This bit indicates the presence of extended APIC register space starting at <a href="#">APIC400</a> .
30:24	Reserved.
23:16	<b>MaxLvtEntry</b> . Read-only. Reset state varies by product. This field specifies the number of entries in the local vector table minus one.
15:8	Reserved.
7:0	<b>Version</b> . Read-only. This field indicates the version number of this APIC implementation.

**APIC80 Task Priority Register**

Reset: 0000 0000h.

Bits	Description
31:8	Reserved.
7:0	<b>Priority</b> . Read-write. This field is assigned by software to set a threshold priority at which the core is interrupted.

**APIC90 Arbitration Priority Register**

Reset: 0000 0000h.

Bits	Description
31:8	Reserved.
7:0	<b>Priority</b> . Read-only. This field indicates the current priority for a pending interrupt, or a task or interrupt being serviced by the core. The priority is used to arbitrate between cores to determine which accepts a lowest-priority interrupt request.

**APICA0 Processor Priority Register**

Reset: 0000 0000h.

Bits	Description



31:8	Reserved.
7:0	<b>Priority.</b> Read-only. This field indicates the core's current priority servicing a task or interrupt, and is used to determine if any pending interrupts should be serviced. It is the higher value of the task priority value and the current highest in-service interrupt.

### APICB0 End of Interrupt Register

This register is written by the software interrupt handler to indicate the servicing of the current interrupt is complete.

Bits	Description
31:0	Reserved. Write only. Reads return undefined data.

### APICC0 Remote Read Register

Reset: 0000 0000h.

Bits	Description
31:0	<b>RemoteReadData.</b> Read-only. This field contains the data resulting from a valid completion of a remote read inter-processor interrupt.

### APICD0 Logical Destination Register

Reset: 0000 0000h.

Bits	Description
31:24	<b>Destination.</b> Read-write. This field contains this APIC's destination identification. This field is used to determine which interrupts should be accepted.
23:0	Reserved.

### APICE0 Destination Format Register

Reset: FFFF FFFFh.

Bits	Description
31:28	<b>Format.</b> Read-write. This field controls which format to use when accepting interrupts with a logical destination mode. The allowed values are: <ul style="list-style-type: none"> <li>• 0h = Cluster destinations are used.</li> <li>• Fh = Flat destinations are used.</li> </ul>
27:0	Reserved.

### APICF0 Spurious Interrupt Vector Register

Reset: 0000 00FFh.

Bits	Description
31:10	Reserved.

9	<b>FocusDisable.</b> Read-write. 1=Disable focus core checking during lowest-priority arbitrated interrupts.
8	<b>APICSWEn: APIC software enable.</b> Read-write. 0=SMI, NMI, INIT, Startup and Remote Read interrupts may be accepted; pending interrupts in <a href="#">APIC[170:100]</a> and <a href="#">APIC[270:200]</a> are held, but further fixed, lowest-priority, LINT, and ExtInt interrupts are not accepted. All LVT entry mask bits are set and cannot be cleared.
7:0	<b>Vector.</b> Read-write. This field contains the vector that is sent to the core in the event of a spurious interrupt. The behavior of bits 3:0 are controlled as specified by <a href="#">[The Link Transaction Control Register] F0x68[ApicExtSpur]</a> .

### APIC[170:100] In-Service Registers

Reset: 0000 0000h. The in-service registers provide a bit per interrupt to indicate that the corresponding interrupt is being serviced by the core. APIC100[15:0] are reserved. Interrupts are mapped as follows:

<u>Register</u>	<u>Interrupt Number</u>
APIC100	31-16
APIC110	63-32
APIC120	95-64
APIC130	127-96
APIC140	159-128
APIC150	191-160
APIC160	223-192
APIC170	255-224

Bits	Description
31:0	<b>InServiceBits.</b> Read-only. These bits are set when the corresponding interrupt is being serviced by the core.

### APIC[1F0:180] Trigger Mode Registers

Reset: 0000 0000h. The trigger mode registers provide a bit per interrupt to indicate the assertion mode of each interrupt. APIC180[15:0] are reserved. Interrupts are mapped as follows:

<u>Register</u>	<u>Interrupt Number</u>
APIC180	31-16
APIC190	63-32
APIC1A0	95-64
APIC1B0	127-96
APIC1C0	159-128
APIC1D0	191-160
APIC1E0	223-192
APIC1F0	255-224

Bits	Description
31:0	<b>TriggerModeBits.</b> Read-only. The corresponding trigger mode bit is updated when an interrupt is accepted. The values are: <ul style="list-style-type: none"> <li>• 0b = edge-triggered interrupt.</li> <li>• 1b = level-triggered interrupt.</li> </ul>

### APIC[270:200] Interrupt Request Registers

Reset: 0000 0000h. The interrupt request registers provide a bit per interrupt to indicate that the corresponding interrupt has been accepted by the APIC. APIC200[15:0] are reserved. Interrupts are mapped as follows:

<u>Register</u>	<u>Interrupt Number</u>
APIC200	31-16
APIC210	63-32
APIC220	95-64
APIC230	127-96
APIC240	159-128
APIC250	191-160
APIC260	223-192
APIC270	255-224

Bits	Description
31:0	<b>RequestBits.</b> Read-only. The corresponding request bit is set when the an interrupt is accepted by the APIC.

### APIC280 Error Status Register

Reset: 0000 0000h. Writes to this register trigger an update of the register state. The value written by software is arbitrary. Each write causes the internal error state to be loaded into this register, clearing the internal error state. Consequently, a second write prior to the occurrence of another error causes the register to be overwritten with cleared data.

Bits	Description
31:8	Reserved.
7	<b>IllegalRegAddr: illegal register address.</b> Read-write. This bit indicates that an access to a non-existent register location within this APIC was attempted.
6	<b>RcvdIllegalVector: received illegal vector.</b> Read-write. This bit indicates that this APIC has received a message with an illegal vector (00h to 0Fh for fixed and lowest priority interrupts).
5	<b>SentIllegalVector.</b> Read-write. This bit indicates that this APIC attempted to send a message with an illegal vector (00h to 0Fh for fixed and lowest priority interrupts).
4	Reserved.
3	<b>RcvAcceptError: receive accept error.</b> Read-write. 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. This bit indicates that a message sent by this APIC was not accepted by any APIC.
1:0	Reserved.

### APIC300 Interrupt Command Register Low

Reset: 0000 0000h. Not all combinations of ICR fields are valid. Only the following combinations are valid:

**Table 105: Valid ICR field 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	Reserved.
19:18	<b>DestShrthnd: destination shorthand.</b> Read-write. This field provides a quick way to specify a destination for a message. The valid encodings are as follows: <ul style="list-style-type: none"> <li>• 00b = Destination field</li> <li>• 01b = Self</li> <li>• 10b = All including self</li> <li>• 11b = All excluding self (Note that 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.)</li> </ul> If all including self or all excluding self is used, then destination mode is ignored and physical is automatically used.
17:16	<b>RemoteRdStat: remote read status.</b> Read-only. The encoding for this field is as follows: <ul style="list-style-type: none"> <li>• 00b = Read was invalid</li> <li>• 01b = Delivery pending</li> <li>• 10b = Delivery done and access was valid</li> <li>• 11b = Reserved</li> </ul>
15	<b>TM: trigger mode.</b> Read-write. This bit indicates how this interrupt is triggered. It is defined as follows: <ul style="list-style-type: none"> <li>• 0 = Edge triggered</li> <li>• 1 = Level triggered</li> </ul>
14	<b>Level.</b> Read-write. The values for this bit are as follows: <ul style="list-style-type: none"> <li>• 0 = Deasserted</li> <li>• 1 = Asserted</li> </ul>
13	Reserved.
12	<b>DlvryStat: delivery status.</b> Read-only. This bit is set to indicate that the interrupt has not yet been accepted by the destination core(s).
11	<b>DM: destination mode.</b> Read-write. The values for this bit are as follows: <ul style="list-style-type: none"> <li>• 0 = Physical</li> <li>• 1 = Logical</li> </ul>

10:8	<b>MsgType.</b> Read-write. The message types are encoded as follows: <ul style="list-style-type: none"> <li>• 000b = Fixed</li> <li>• 001b = Lowest Priority</li> <li>• 010b = SMI</li> <li>• 011b = Remote read</li> <li>• 100b = NMI</li> <li>• 101b = INIT</li> <li>• 110b = Startup</li> <li>• 111b = External interrupt</li> </ul>
7:0	<b>Vector.</b> Read-write. This field contains the vector that is sent for this interrupt source.

### APIC310 Interrupt Command Register High

Reset: 0000 0000h.

Bits	Description
31:24	<b>DestinationField.</b> Read-write. This field contains the destination encoding used when <a href="#">APIC300[DestShrthnd]</a> is 00b.
23:0	Reserved.

### APIC320 Timer Local Vector Table Entry

Reset: 0001 0000h.

Bits	Description
31:18	Reserved.
17	<b>Mode.</b> Read-write. The values for this bit are as follows: <ul style="list-style-type: none"> <li>• 0 = One-shot</li> <li>• 1 = Periodic</li> </ul>
16	<b>Mask.</b> Read-write. If this bit is set, this local vector table entry does not generate interrupts.
15:13	Reserved.
12	<b>DlvryStat: delivery status.</b> Read-only. This bit is set to indicate that the interrupt has not yet been accepted by the core.
11	Reserved.
10:8	<b>MsgType: message type.</b> Write only. Read always returns 000b. See <a href="#">2.13.1.10 [Generalized Local Vector Table]</a> for supported message types.
7:0	<b>Vector.</b> Read-write. This field contains the vector that is sent for this interrupt source.

### APIC330 Thermal Local Vector Table Entry

Reset: 0001 0000h. Interrupts for this local vector table are caused by transitions in and out of an STC thermal zone as described in [2.10.3.2 \[Software Thermal Control \(STC\)\]](#), changes in [\[The P-State Current Limit Register\] MSRC001\\_0061\[CurPstateLimit\]](#) due to SB-RMI, HTC, or STC.

Bits	Description
31:17	Reserved.
16	<b>Mask.</b> Read-write. If this bit is set, this local vector table entry does not generate interrupts.

15:13	Reserved.
12	<b>DlvryStat: delivery status.</b> Read-only. This bit is set to indicate that the interrupt has not yet been accepted by the core.
11	Reserved.
10:8	<b>MsgType: message type.</b> Read-write. See 2.13.1.10 [Generalized Local Vector Table] for supported message types.
7:0	<b>Vector.</b> Read-write. This field contains the vector that is sent for this interrupt source.

### APIC340 Performance Counter Vector Table Entry

Reset: 0001 0000h. Interrupts for this local vector table are caused by overflows of [The Performance Event Counter Registers (PERF\_CTR[3:0])] MSRC001\_00[07:04]. Note: The Mask bit is not set automatically when the interrupt is taken.

Bits	Description
31:17	Reserved.
16	<b>Mask.</b> Read-write. If this bit is set, this local vector table entry does not generate interrupts.
15:13	Reserved.
12	<b>DlvryStat: delivery status.</b> Read-only. This bit is set to indicate that the interrupt has not yet been accepted by the core.
11	Reserved.
10:8	<b>MsgType: message type.</b> Read-write. See 2.13.1.10 [Generalized Local Vector Table] for supported message types.
7:0	<b>Vector.</b> Read-write. This field contains the vector that is sent for this interrupt source.

### APIC350 Local Interrupt 0 (Legacy INTR) Local Vector Table Entry

Reset: 0001 0000h.

Bits	Description
31:17	Reserved.
16	<b>Mask.</b> Read-write. If this bit is set, this local vector table entry does not generate interrupts.
15	<b>TM: trigger mode.</b> Read-write. This bit indicates how this interrupt is triggered. It is defined as follows: <ul style="list-style-type: none"> <li>• 0 = Edge triggered</li> <li>• 1 = Level triggered</li> </ul>
14	<b>RmtIRR.</b> Read-only. If trigger mode is level, remote IRR is set when the interrupt has begun service. Remote IRR is cleared when the end of interrupt has occurred.
13	<b>PinPol: pin polarity.</b> Read-write. This bit is not used because LINT interrupts are delivered by HyperTransport™ messages instead of individual pins.
12	<b>DlvryStat: delivery status.</b> Read-only. This bit is set to indicate that the interrupt has not yet been accepted by the core.
11	Reserved.
10:8	<b>MsgType: message type.</b> Read-write. See 2.13.1.10 [Generalized Local Vector Table] for supported message types.

7:0	<b>Vector.</b> Read-write. This field contains the vector that is sent for this interrupt source.
-----	---

### APIC360 Local Interrupt 1(Legacy NMI) Local Vector Table Entry

Reset: 0001 0000h.

Bits	Description
31:0	See <a href="#">APIC350</a> .

### APIC370 Error Local Vector Table Entry

Reset: 0001 0000h.

Bits	Description
31:17	Reserved.
16	<b>Mask.</b> Read-write. If this bit is set, this local vector table entry does not generate interrupts.
15:13	Reserved.
12	<b>DelvryStat: delivery status.</b> Read only. This bit is set to indicate that the interrupt has not yet been accepted by the core.
11	Reserved.
10:8	<b>MsgType: message type.</b> Write only. Read always returns 0h. See <a href="#">2.13.1.10 [Generalized Local Vector Table]</a> for supported message types.
7:0	<b>Vector.</b> Read-write. This field contains the vector that is sent for this interrupt source.

### APIC380 Timer Initial Count Register

Reset: 0000 0000h.

Bits	Description
31:0	<b>Count.</b> Read-write. This field contains the value copied into the current count register when the timer is loaded or reloaded.

### APIC390 Timer Current Count Register

Reset: 0000 0000h.

Bits	Description
31:0	<b>Count.</b> Read only. This field contains the current value of the counter.

### APIC3E0 Timer Divide Configuration Register

Reset: 0000 0000h. The Div bits are encoded as follows:

Div[3]	Div[1:0]	Resulting Timer Divide
0	00b	2
0	01b	4
0	10b	8



Div[3]	Div[1:0]	Resulting Timer Divide
0	11b	16
1	00b	32
1	01b	64
1	10b	128
1	11b	1

Bits	Description
31:4	Reserved.
3	<b>Div[3]</b> . Read-write.
2	Reserved.
1:0	<b>Div[1:0]</b> . Read-write.

### APIC400 Extended APIC Feature Register

Bits	Description
31:24	Reserved.
23:16	<b>ExtLvtCount: extended local vector table count</b> . Read-only, 04h. This specifies the number of extended LVT registers in the local APIC. These registers are <a href="#">[The Extended Interrupt [3:0] Local Vector Table Registers] APIC[530:500]</a> .
15:3	Reserved.
2	<b>ExtApicIdCap: extended APIC ID capable</b> . Read-only, 1. Indicates that the processor is capable of supporting an 8-bit APIC ID, controlled by <a href="#">APIC410[ExtApicIdEn]</a> .
1	<b>SeioCap: specific end of interrupt capable</b> . Read-only, 1. This bit indicates that the <a href="#">[The Specific End Of Interrupt Register] APIC420</a> is present.
0	<b>IerCap: interrupt enable register capable</b> . Read-only, 1. This bit indicates that the <a href="#">[The Interrupt Enable Registers] APIC[4F0:480]</a> are present. See 2.13.1.5 <a href="#">[Interrupt Masking]</a> .

### APIC410 Extended APIC Control Register

Reset: 0000 0000h.

Bits	Description
31:3	Reserved.
2	<b>ExtApicIdEn: extended APIC ID enable</b> . Read-write. 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 $\text{IntDest}[7:0]=1111\_1111$ (instead of $\text{xxxx\_1111}$ ); a match in physical destination mode occurs when $(\text{IntDest}[7:0] == \text{ApicId}[7:0])$ instead of $(\text{IntDest}[3:0] == \text{ApicId}[3:0])$ . Extended APIC ID can also be enabled by writing <a href="#">F0x68[ApicExtId]</a> and <a href="#">F0x68[ApicExtBrdCst]</a> . If <a href="#">ExtApicIdEn=1</a> , <a href="#">F0x68[ApicExtId]</a> and <a href="#">F0x68[ApicExtBrdCst]</a> must also be 1.
1	<b>SeoiEn</b> . Read-write. This bit enables SEOI generation when a write to the specific end of interrupt register is received.
0	<b>IerEn</b> . Read-write. This bit enables writes to the interrupt enable registers.

**APIC420 Specific End Of Interrupt Register**

Reset: 0000 0000h

Bits	Description
31:8	Reserved.
7:0	<b>EoiVec: end of interrupt vector.</b> Read-write. A write to this field causes an end of interrupt cycle to be performed for the vector specified in this field. The behavior is undefined if no interrupt is pending for the specified interrupt vector.

**APIC[4F0:480] Interrupt Enable Registers**

Reset: FFFF FFFFh

Bits	Description																		
31:0	<b>InterruptEnableBits.</b> Read-write. The interrupt enable bits can be used to enable each of the 256 interrupts. Interrupt enables are mapped as follows: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Register</th> <th>Interrupt Number</th> </tr> </thead> <tbody> <tr> <td>APIC480</td> <td>31-0</td> </tr> <tr> <td>APIC490</td> <td>63-32</td> </tr> <tr> <td>APIC4A0</td> <td>95-64</td> </tr> <tr> <td>APIC4B0</td> <td>127-96</td> </tr> <tr> <td>APIC4C0</td> <td>159-128</td> </tr> <tr> <td>APIC4D0</td> <td>191-160</td> </tr> <tr> <td>APIC4E0</td> <td>223-192</td> </tr> <tr> <td>APIC4F0</td> <td>255-224</td> </tr> </tbody> </table>	Register	Interrupt Number	APIC480	31-0	APIC490	63-32	APIC4A0	95-64	APIC4B0	127-96	APIC4C0	159-128	APIC4D0	191-160	APIC4E0	223-192	APIC4F0	255-224
Register	Interrupt Number																		
APIC480	31-0																		
APIC490	63-32																		
APIC4A0	95-64																		
APIC4B0	127-96																		
APIC4C0	159-128																		
APIC4D0	191-160																		
APIC4E0	223-192																		
APIC4F0	255-224																		

**APIC[530:500] Extended Interrupt [3:0] Local Vector Table Registers**

Reset: 0001 0000h. These registers provide additional local vector table entries for selected internal interrupt sources, including those found in: [F3x1\[78, 70, 68, 60\]](#) and [F3xB0](#).

Bits	Description
31:17	Reserved.
16	<b>Mask.</b> Read-write. 1=This LVT entry does not generate interrupts.
15:13	Reserved.
12	<b>DlvryStat: delivery status.</b> Read-only. 1=The interrupt has not yet been accepted by the CPU.
11	Reserved.
10:8	<b>MsgType: message type.</b> Read-write. Specifies the interrupt type generated by this LVT entry. See <a href="#">2.13.1.10 [Generalized Local Vector Table]</a> for supported message types.
7:0	<b>Vector.</b> Read-write. This field contains the vector generated by this LVT entry.

**3.9 CPUID Instruction Registers**

Processor feature capabilities and configuration information are provided through the CPUID instruction. Different information is accessed by (1) setting EAX as an index to the registers to be read, (2) executing the CPUID instruction, and (3) reading the results in EAX, EBX, ECX, and EDX. The phrase *CPUID function X* or *CPUID FnX* refers to the CPUID instruction when EAX is preloaded with X. Undefined function numbers

return 0's in all 4 registers. See [2.15 \[CPUID Instruction\]](#) also.

The following provides AMD family 10h processor specific details about CPUID. See the *CPUID Specification* for further information. 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.

### **CPUID Fn[8000\_0000, 0000\_0000] AMD Authentic Identifier**

Register	Bits	Description
EAX	31:0	<b>LFuncStd: largest standard function.</b> Function 0000_0000h returns the largest CPUID standard-function input value supported by the processor implementation: Revision D and earlier: 0000_0005h. Revision E: 0000_0006h. <b>LFuncExt: largest extended function.</b> Function 8000_0000h returns the largest CPUID extended-function input value supported by the processor implementation: Revision B: 8000_001Ah. Revision C: 8000_001Bh.
EBX, ECX, EDX	31:0	<b>Vendor: vendor.</b> The 12 8-bit ASCII character codes to create the string “AuthenticAMD”. EBX=6874_7541h “h t u A”, ECX=444D_4163h “D M A c”, EDX=6974_6E65h “i t n e”.

### **CPUID Fn[8000\_0001, 0000\_0001]\_EAX Family, Model, Stepping Identifiers**

This register provides identical information to [F3xFC](#).

**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]=01h, then Family[7:0]=10h. This document applies only to family 10h processors.

**Model** is an 8-bit value and is defined as: **Model[7:0]** = {ExtendedModel[3:0], BaseModel[3:0]}. E.g. If ExtendedModel[3:0]=Eh and BaseModel[3:0]=8h, then Model[7:0] = E8h. Model numbers vary with product.

Bits	Description
31:28	Reserved.
27:20	<b>ExtendedFamily:</b> 01h.
19:16	<b>ExtendedModel.</b>
15:12	Reserved.
11:8	<b>BaseFamily:</b> Fh.
7:4	<b>BaseModel.</b>
3:0	<b>Stepping:</b> processor stepping (revision) for a specific model.

**CPUID Fn0000\_0001\_EBX LocalApicId, LogicalProcessorCount, CLFlush, 8BitBrandId**

Bits	Description
31:24	<b>LocalApicId:</b> initial local APIC physical ID. Provides the initial APIC20[ApicId] value. After F0x60[NodeId] as been initialized, changes to APIC20[ApicId] do not affect the value of this CPUID register. See 2.9.2 [CPU Cores and Downcoring].
23:16	<b>LogicalProcessorCount:</b> If CPUID Fn[8000_0001, 0000_0001]_EDX[HTT] = 1, then this field indicates the number of cores in the processor as CPUID Fn8000_0008[NC] + 1. Otherwise, this field is reserved.
15:8	<b>CLFlush:</b> CLFLUSH size in quadwords = 08h.
7:0	<b>8BitBrandId:</b> 8 bit brand ID = 00h. Indicates that the brand ID is in CPUID Fn8000_0001_EBX.

**CPUID Fn8000\_0001\_EBX BrandId Identifier**

Bits	Description
31:28	<b>PkgType:</b> package type. Specifies the processor package type. This field is encoded as follows: 0000b: Fr2(1207), Fr5(1207), or Fr6(1207). 0001b: AM2r2 or AM3. 0010b: S1g3 or S1g4. 0011b: G34. 0100b: ASB2. 0101b: C32. All other encodings are reserved.
27:16	Reserved.
15:0	<b>BrandId:</b> brand ID. This is identical to F3x1F0[BrandId].

**CPUID Fn0000\_0001\_ECX Feature Identifiers**

Bits	Description
31	RAZ. Reserved for use by hypervisor to indicate guest status.
30:24	Reserved.
23	<b>POPCNT:</b> POPCNT instruction = 1.
22:14	Reserved.
13	<b>CMPXCHG16B:</b> CMPXCHG16B instruction = 1.
12:4	Reserved.
3	<b>Monitor:</b> Monitor/Mwait instructions = 1. This can be disabled through [The Hardware Configuration Register (HWCR)] MSRC001_0015[MonMwaitDis].
2:1	Reserved.
0	<b>SSE3:</b> SSE3 extensions = 1; may be overridden by MSRC001_0015[SseDis].

**CPUID Fn8000\_0001\_ECX Feature Identifiers**

Bits	Description
31:20	Reserved
19	<b>NodeId</b> : Indicates support for <a href="#">MSRC001_100C</a> [NodeId, NodesPerProcessor].
18:14	Reserved.
13	<b>WDT</b> : watchdog timer support. Revision B = 0. Revision C = 1.
12	<b>SKINIT</b> : SKINIT and STGI support. Revision B = 0. Revision C = 1.
11	Reserved.
10	<b>IBS</b> : Instruction Based Sampling = 1.
9	<b>OSVW</b> : OS Visible Work-around support = 1.
8	<b>3DNowPrefetch</b> : Prefetch and PrefetchW instructions = 1.
7	<b>MisAlignSse</b> : Misaligned SSE Mode = (setting varies by product); may be overridden by <a href="#">MSRC001_0015</a> [MisAlignSseDis].
6	<b>SSE4A</b> : EXTRQ, INSERTQ, MOVNTSS, and MOVNTSD instruction support = 1; may be overridden by <a href="#">MSRC001_0015</a> [SseDis].
5	<b>ABM</b> : advanced bit manipulation. LZCNT instruction support (setting varies by product).
4	<b>AltMovCr8</b> : LOCK MOV CR0 means MOV CR8 = 1.
3	<b>ExtApicSpace</b> : extended APIC register space = 1.
2	<b>SVM</b> : Secure Virtual Mode feature (setting varies by product).
1	<b>CmpLegacy</b> : core multi-processing legacy mode (setting varies by product). 1=Multi core product ( <a href="#">CPUID Fn8000_0008</a> [NC] != 0). 0=Single core product ( <a href="#">CPUID Fn8000_0008</a> [NC] = 0).
0	<b>LahfSahf</b> : LAHF/SAHF instructions = 1.

**CPUID Fn[8000\_0001, 0000\_0001]\_EDX Feature Identifiers**

The value returned in EDX may be identical or different for Fn0000\_0001 and Fn8000\_0001, as indicated.

Bits	Function	Description
31	0000_0001h	Reserved.
	8000_0001h	<b>3DNow</b> : 3DNow! <sup>TM</sup> instructions = 1.
30	0000_0001h	Reserved.
	8000_0001h	<b>3DNowExt</b> : AMD extensions to 3DNow! <sup>TM</sup> instructions = 1.
29	0000_0001h	Reserved.
	8000_0001h	<b>LM</b> : long mode (may vary by product).
28	0000_0001h	<b>HTT</b> : hyper-threading technology (setting varies by product). This bit qualifies the meaning of <a href="#">CPUID Fn0000_0001_EBX</a> [LogicalProcessorCount]. 1=Multi core product ( <a href="#">CPUID Fn8000_0008</a> [NC] != 0). 0=Single core product ( <a href="#">CPUID Fn8000_0008</a> [NC] = 0).
	8000_0001h	Reserved.

Bits	Function	Description
27	0000_0001h	Reserved.
	8000_0001h	<b>RDTSCP</b> : RDTSCP instruction = 1.
26	0000_0001h	<b>SSE2</b> : SSE2 extensions = 1; may be overridden by <a href="#">MSRC001_0015[SseDis]</a> .
	8000_0001h	<b>Page1GB</b> : 1 GB large page support = 1.
25	0000_0001h	<b>SSE</b> : SSE extensions = 1; may be overridden by <a href="#">MSRC001_0015[SseDis]</a> .
	8000_0001h	<b>FXSR</b> : FXSAVE and FXRSTOR instruction optimizations = 1.
24	both	<b>FXSR</b> : FXSAVE and FXRSTOR instructions = 1.
23	both	<b>MMX</b> : MMX™ instructions = 1.
22	0000_0001h	Reserved.
	8000_0001h	<b>MmxExt</b> : AMD extensions to MMX™ instructions = 1.
21	Both	Reserved.
20	0000_0001h	Reserved.
	8000_0001h	<b>NX</b> : no-execute page protection = 1.
19	0000_0001h	<b>CLFSH</b> : CLFLUSH instruction = 1.
	8000_0001h	Reserved.
18	Both	Reserved.
17	both	<b>PSE36</b> : page-size extensions = 1.
16	both	<b>PAT</b> : page attribute table = 1.
15	both	<b>CMOV</b> : conditional move instructions, CMOV, FCOMI, FCMOV = 1.
14	both	<b>MCA</b> : machine check architecture, MCG_CAP = 1.
13	both	<b>PGE</b> : page global extension, CR4.PGE = 1.
12	both	<b>MTRR</b> : memory-type range registers = 1.
11	0000_0001h	<b>SysEnterSysExit</b> : SYSENTER and SYSEXIT instructions = 1.
	8000_0001h	<b>SysCallSysRet</b> : SYSCALL and SYSRET instructions = 1.
10	both	Reserved.
9	both	<b>APIC</b> : advanced programmable interrupt controller (APIC) exists and is enabled. This bit reflects the state of <a href="#">[The APIC Base Address Register (APIC_BAR)]</a> <a href="#">MSR0000_001B[ApicEn]</a> .
8	both	<b>CMPXCHG8B</b> : CMPXCHG8B instruction = 1.
7	both	<b>MCE</b> : machine check exception, CR4.MCE = 1.
6	both	<b>PAE</b> : physical-address extensions (PAE) = 1.
5	both	<b>MSR</b> : AMD model-specific registers (MSRs), with RDMSR and WRMSR instructions = 1.
4	both	<b>TSC</b> : time stamp counter, RDTSC/RDTSCP instructions, CR4.TSD = 1.
3	both	<b>PSE</b> : page-size extensions (4 MB pages) = 1.
2	both	<b>DE</b> : debugging extensions, IO breakpoints, CR4.DE = 1.
1	both	<b>VME</b> : virtual-mode enhancements = 1.
0	both	<b>FPU</b> : x87 floating point unit on-chip = 1.

**CPUID Fn0000\_000[4, 3, 2] Reserved****CPUID Fn8000\_000[4, 3, 2] Processor Name String Identifier**

These return the ASCII string corresponding to the processor name, stored in [The Processor Name String Registers] MSRC001\_00[35:30]. The MSRs are mapped to these registers as follows:

Function 8000\_0002: {EDX, ECX, EBX, EAX} == {MSRC001\_0031, MSRC001\_0030};

Function 8000\_0003: {EDX, ECX, EBX, EAX} == {MSRC001\_0033, MSRC001\_0032};

Function 8000\_0004: {EDX, ECX, EBX, EAX} == {MSRC001\_0035, MSRC001\_0034};

**CPUID Fn0000\_0005 Monitor/MWait**

Register	Bits	Description
EAX	31:16	Reserved.
EAX	15:0	Smallest monitor-line size in bytes = 40h.
EBX	31:16	Reserved.
EBX	15:0	Largest monitor-line size in bytes = 40h.
ECX	31:2	Reserved.
ECX	1	<b>IBE</b> : Interrupt break-event = 1.
ECX	0	<b>EMX</b> : Enumerate MONITOR/MWAIT extensions = 1.
EDX	31:0	Reserved.

**CPUID Fn8000\_0005 TLB and L1 Cache Identifiers**

This provides the processor's first level cache and TLB characteristics for each core. The *associativity* fields returned are encoded as follows:

00h Reserved.

01h Direct mapped.

02h - FEh Specifies the associativity; e.g., 04h would indicate a 4-way associativity.

FFh Fully associative

Register	Bits	Description
EAX	31:24	Data TLB associativity for 2 MB and 4 MB pages = FFh.
EAX	23:16	Data TLB number of entries for 2 MB and 4 MB pages = 48. The value returned is for the number of entries available for the 2 MB page size; 4 MB pages require two 2 MB entries, so the number of entries available for the 4 MB page size is one-half the returned value.
EAX	15:8	Instruction TLB associativity for 2 MB and 4 MB pages = FFh.
EAX	7:0	Instruction TLB number of entries for 2 MB and 4 MB pages = 16. The value returned is for the number of entries available for the 2 MB page size; 4 MB pages require two 2 MB entries, so the number of entries available for the 4 MB page size is one-half the returned value.
EBX	31:24	Data TLB associativity for 4 KB pages = FFh.
EBX	23:16	Data TLB number of entries for 4 KB pages = 48.
EBX	15:8	Instruction TLB associativity for 4 KB pages = FFh.



Register	Bits	Description
EBX	7:0	Instruction TLB number of entries for 4 KB pages = 32.
ECX	31:24	L1 data cache size in KB = 64.
ECX	23:16	L1 data cache associativity = 2.
ECX	15:8	L1 data cache lines per tag = 1.
ECX	7:0	L1 data cache line size in bytes = 64.
EDX	31:24	L1 instruction cache size KB = 64.
EDX	23:16	L1 instruction cache associativity = 2.
EDX	15:8	L1 instruction cache lines per tag = 1.
EDX	7:0	L1 instruction cache line size in bytes = 64.

### CPUID Fn0000\_0006 Power Management

Register	Bits	Description
EAX	31:0	Reserved.
EBX	31:0	Reserved.
ECX	31:1	Reserved.
ECX	0	<b>EffFreq: effective frequency interface.</b> Revision D and earlier: 0. Revision E: 1. 1=Indicates presence of [The Max Performance Frequency Clock Count (MPERF)] MSR0000_00E7 and [The Actual Performance Frequency Clock Count (APERF)] MSR0000_00E8.
EDX	31:0	Reserved.

### CPUID Fn8000\_0006 L2/L3 Cache and L2 TLB Identifiers

This provides the processor's second level cache and TLB characteristics for each core and the processor's third level cache characteristics shared by all cores.

The presence of a unified L2 TLB is indicated by a value of 0000h in the upper 16 bits of the EAX and EBX registers. The unified L2 TLB information is contained in the lower 16 bits of these registers.

The *associativity* fields are encoded as follows:

0h: The L2 cache or TLB is disabled.

1h: Direct mapped.

2h: 2-way associative.

4h: 4-way associative.

6h: 8-way associative.

8h: 16-way associative.

Ah: 32-way associative.

Bh: 48-way associative.

Ch: 64-way associative.

Dh: 96-way associative.

Eh: 128-way associative.

Fh: Fully associative.

All other encodings are reserved.

Register	Bits	Description
EAX	31:28	<b>L2DTlb2and4MAssoc.</b> L2 data TLB associativity for 2 MB and 4 MB pages = 2.
EAX	27:16	<b>L2DTlb2and4MSize.</b> L2 data TLB number of entries for 2 MB and 4 MB pages = 128. The value returned is for the number of entries available for the 2 MB page size; 4 MB pages require two 2 MB entries, so the number of entries available for the 4 MB page size is one-half the returned value.
EAX	15:12	<b>L2ITlb2and4MAssoc.</b> L2 instruction TLB associativity for 2 MB and 4 MB pages = 0.
EAX	11:0	<b>L2ITlb2and4MSize.</b> L2 instruction TLB number of entries for 2 MB and 4 MB pages = 0.
EBX	31:28	<b>L2DTlb4KAssoc.</b> L2 data TLB associativity for 4 KB pages = 4.
EBX	27:16	<b>L2DTlb4KSize.</b> L2 data TLB number of entries for 4 KB pages = 512.
EBX	15:12	<b>L2ITlb4KAssoc.</b> L2 instruction TLB associativity for 4 KB pages = 4.
EBX	11:0	<b>L2ITlb4KSize.</b> L2 instruction TLB number of entries for 4 KB pages = 512.
ECX	31:16	<b>L2Size.</b> L2 cache size in KB (varies with product). May be one of 512 or 1024.
ECX	15:12	<b>L2Assoc.</b> L2 cache associativity = 8.
ECX	11:8	<b>L2LinesPerTag.</b> L2 cache lines per tag = 1. $13 * 2 - \text{rounded}(\text{pfsiz} * 2)$
ECX	7:0	<b>L2LineSize.</b> L2 cache line size in bytes = 64.
EDX	31:18	<b>L3Size.</b> L3 cache size (varies with product). L3 cache size for the processor is at least $(\text{L3Size}[31:18] * 512\text{KB})$ and less than $((\text{L3Size}[31:18] + 1) * 512\text{KB})$ . This does not include any amount consumed by the probe filter. See 2.6.5.1 [Probe Filter].
EDX	17:16	Reserved.
EDX	15:12	<b>L3Assoc.</b> L3 cache associativity = (varies with product); supported values are 16, 32, 48, 64, 96, and 128. When probe filter is enabled, the associativity available for L3 cache data is smaller than the value reported. See 2.6.5.1 [Probe Filter].
EDX	11:8	<b>L3LinesPerTag.</b> L3 cache lines per tag = 1.
EDX	7:0	<b>L3LineSize.</b> L3 cache line size in bytes = 64.

### CPUID Fn8000\_0007 Advanced Power Management Information

This function provides advanced power management feature identifiers.

Register	Bits	Description
EAX, EBX, ECX	31:0	Reserved.
EDX	31:10	Reserved.
EDX	9	<b>CPB.</b> Core performance boost. Revision D and earlier = 0. Revision E: Support varies by product.
EDX	8	<b>TscInvariant:</b> TSC rate is invariant = 1.
EDX	7	<b>Hwpstate:</b> hardware P-state control is supported = 1. [The P-State Current Limit Register] MSRC001_0061, [The P-State Control Register] MSRC001_0062 and [The P-State Status Register] MSRC001_0063 exist.
EDX	6	<b>100MHzSteps:</b> 100 MHz multiplier Control = 1.
EDX	5	<b>STC:</b> software thermal control (STC) is supported (support may vary by product).

Register	Bits	Description
EDX	4	<b>TM:</b> hardware thermal control (HTC) is supported (support may vary by product).
EDX	3	<b>TTP:</b> THERMTRIP is supported = 1.
EDX	2	<b>VID:</b> Voltage ID control is supported = 0 (function replaced by HwPstate).
EDX	1	<b>FID:</b> Frequency ID control is supported = 0 (function replaced by HwPstate).
EDX	0	<b>TS:</b> Temperature sensor = 1.

### CPUID Fn8000\_0008 Address Size And Physical Core Count Information

This provides information about the number of physical cores and the maximum physical and linear address width supported by the processor.

Register	Bits	Description
EAX	31:16	Reserved.
EAX	15:8	Maximum linear byte address size in bits. If the processor supports long mode (see <a href="#">CPUID Fn[8000_0001, 0000_0001]_EDX[LM]</a> ) then this is 30h; else this is 20h.
EAX	7:0	Maximum physical byte address size in bits = 30h.
EBX	31:0	Reserved.
ECX	31:16	Reserved.
ECX	15:12	<b>ApicIdCoreIdSize[3:0].</b> The number of bits in the initial <a href="#">APIC20[ApicId]</a> value that indicate core ID within a processor = 2h, 3h, or 4h.
ECX	11:8	Reserved.
ECX	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). This value is affected by <a href="#">F3x190[DisCore]</a> . See <a href="#">2.9.2 [CPU Cores and Downcoring]</a> .
EDX	31:0	Reserved.

### CPUID Fn8000\_0009 Reserved

### CPUID Fn8000\_000A SVM Revision and Feature Identification

This provides SVM revision and feature information. If [CPUID Fn8000\\_0001\\_ECX\[SVM\]](#)=0 then [CPUID Fn8000\\_000A](#) is reserved.

Register	Bits	Description
EAX	31:8	Reserved.
EAX	7:0	<b>SvmRev:</b> SVM revision = 01h.
EBX	31:0	<b>NASID:</b> number of address space identifiers (ASID) = 40h.
ECX	31:0	Reserved.
EDX	31:11	Reserved.
EDX	10	<b>PauseFilter:</b> Pause intercept filter. Revision C and earlier = 0. Revision D and later = 1.
EDX	9:4	Reserved.
EDX	3	<b>NRIPS:</b> NRIP Save. Revision B = 0. Revision C = 1.

Register	Bits	Description
EDX	2	<b>SVML</b> : SVM lock = 1.
EDX	1	<b>LbrVirt</b> : LBR virtualization = 1.
EDX	0	<b>NP</b> : Nested Paging = 1.

### CPUID Fn8000\_00[18:0B] Reserved

---

### CPUID Fn8000\_0019 TLB 1GB Page Identifiers

---

This provides 1 GB paging information. The *associativity* fields are defined by [CPUID Fn8000\\_0006](#).

Register	Bits	Description
EAX	31:28	L1 data TLB associativity for 1 GB pages = Fh.
EAX	27:16	L1 data TLB number of entries for 1 GB pages = 48.
EAX	15:12	L1 instruction TLB associativity for 1 GB pages = 0.
EAX	11:0	L1 instruction TLB number of entries for 1 GB pages = 0.
EBX	31:28	L2 data TLB associativity for 1 GB pages. Revision B = 0. Revision C = 6.
EBX	27:16	L2 data TLB number of entries for 1 GB pages. Revision B = 0. Revision C = 16.
EBX	15:12	L2 instruction TLB associativity for 1 GB pages = 0.
EBX	11:0	L2 instruction TLB number of entries for 1 GB pages = 0.
ECX	31:0	Reserved.
EDX	31:0	Reserved.

### CPUID Fn8000\_001A Performance Optimization Identifiers

---

This function returns performance related information.

Register	Bits	Description
EAX	31:2	Reserved.
EAX	1	<b>MOVU</b> .
EAX	0	<b>FP128</b> .
EBX	31:0	Reserved.
ECX	31:0	Reserved.
EDX	31:0	Reserved.

### CPUID Fn8000\_001B Instruction Based Sampling Identifiers

---

This function returns IBS feature information.

Register	Bits	Description
EAX	31:6	Reserved.
EAX	5	<b>BrnTrgt</b> . branch target address reporting supported = 0.
EAX	4	<b>OpCnt</b> . op counting mode supported. Revision B = 0. Revision C = 1.

Register	Bits	Description
EAX	3	<b>RdWrOpCnt.</b> read write of op counter supported. Revision B = 0. Revision C = 1.
EAX	2	<b>OpSam.</b> IBS execution sampling supported. Revision B = 0. Revision C = 1.
EAX	1	<b>FetchSam.</b> IBS fetch sampling supported. Revision B = 0. Revision C = 1.
EAX	0	<b>IBSFFV.</b> IBS feature flags valid. Revision B = 0. Revision C = 1.
EBX	31:0	Reserved.
ECX	31:0	Reserved.
EDX	31:0	Reserved.

### 3.10 MSRs - MSR0000\_xxxx

See 3.1 [Register Descriptions and Mnemonics] for a description of the register naming convention. MSRs are accessed through x86 WRMSR and RDMSR instructions.

#### **MSR0000\_0000 Load-Store MCA Address Register**

This is an alias of [MSR0000\\_040E](#).

#### **MSR0000\_0001 Load-Store MCA Status Register**

This is an alias of [MSR0000\\_040D](#).

#### **MSR0000\_0010 Time Stamp Counter Register (TSC)**

Reset: 0000 0000 0000 0000h.

Bits	Description
63:0	<b>TSC: time stamp counter.</b> Read-write. After reset, this register increments by one for each clock cycle. The TSC counts at the same rate in all P-states, all C states, S0, or S1. See <a href="#">MSRC001_0015</a> [Tsc-FreqSel].

#### **MSR0000\_001B APIC Base Address Register (APIC\_BAR)**

Reset: 0000 0000 FEE0 0?00h; bits[11:9] reset to 000b; see below for bit[8].

Bits	Description
63:48	MBZ.
47:12	<b>ApicBar: APIC base address register.</b> Read-write. Specifies the base address for the APICXX register set. See 3.8 [APIC Registers] for details about this register set.
11	<b>ApicEn: APIC enable.</b> Read-write. 1=Local APIC enabled; the APICXX register set is accessible; all interrupt types are accepted. 0=Local APIC disabled; the APICXX register set is not accessible; only non-vectored interrupts are supported including NMI, SMI, INIT and ExtINT; local-vector-table interrupts can still occur if the LVTs have been previously programmed.
10:9	MBZ.
8	<b>BSC: boot strap core.</b> Read-write. 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 Register (EBL\_CR\_POWERON)**

Reset: 0000 0000 0000 0000h. Attempted writes to this register result in general protection faults with error code 0.

Bits	Description
63:18	Reserved.
17:16	<b>ClusterID</b> . Read-only. This is normally 00b; the value does not affect hardware.
15:0	Reserved.

**MSR0000\_00E7 Max Performance Frequency Clock Count (MPERF)**

Revision D and earlier: Reserved. Revision E: Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	<b>MPERF: maximum core clocks counter</b> . Read-write. Incremented by hardware at the P0 frequency while the core is in the C0 state. In combination with <a href="#">MSR0000_00E8</a> , this is used to determine the effective frequency of the core. This field uses software P-state numbering. See <a href="#">MSRC001_0015</a> [Eff-FreqCntMwait], <a href="#">2.4.5</a> [Effective Frequency], and <a href="#">2.4.2.1.2.1</a> [Software P-state Numbering].

**MSR0000\_00E8 Actual Performance Frequency Clock Count (APERF)**

Revision D and earlier: Reserved. Revision E: Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:0	<b>APERF: actual core clocks counter</b> . Read-write. This register increments in proportion to the actual number of core clock cycles while the core is in the C0 state. See <a href="#">MSR0000_00E7</a> .

**MSR0000\_00FE MTRR Capabilities Register (MTRRcap)**

Reset: 0000 0000 0000 0508h.

Bits	Description
63:11	Reserved.
10	<b>MtrrCapWc: write-combining memory type</b> . Read-only. 1=The write combining memory type is supported.
9	Reserved.
8	<b>MtrrCapFix: fixed range register</b> . Read-only. 1=Fixed MTRRs are supported.
7:0	<b>MtrrCapVCnt: variable range registers count</b> . Read-only. Specifies the number of variable MTRRs supported.

**MSR0000\_0174 SYSENTER CS Register (SYSENTER\_CS)**

Reset: 0000 0000 0000 0000h.

Bits	Description
63:32	RAZ.
31:16	Reserved.
15:0	<b>SYSENTER_CS: SYSENTER target CS.</b> Read-write. Holds the called procedure code segment.

#### **MSR0000\_0175 SYSENTER ESP Register (SYSENTER\_ESP)**

Reset: 0000 0000 0000 0000h.

Bits	Description
63:32	Reserved.
31:0	<b>SYSENTER_ESP: SYSENTER target SP.</b> Read-write. Holds the called procedure stack pointer.

#### **MSR0000\_0176 SYSENTER EIP Register (SYSENTER\_EIP)**

Reset: 0000 0000 0000 0000h.

Bits	Description
63:32	Reserved.
31:0	<b>SYSENTER_EIP: SYSENTER target IP.</b> Read-write. Holds the called procedure instruction pointer.

#### **MSR0000\_0179 Global Machine Check Capabilities Register (MCG\_CAP)**

Reset: 0000 0000 0000 0106h.

Bits	Description
63:9	Reserved
8	<b>MCG_CTL_P: MCG_CTL register present.</b> Read-only. 1=The machine check control registers (MCI_CTL; see <a href="#">2.12.1 [Machine Check Architecture]</a> ) are present.
7:0	<b>Count.</b> Read-only. Indicates the number of error-reporting banks visible to each core.

#### **MSR0000\_017A Global Machine Check Status Register (MCG\_STAT)**

Reset: 0000 0000 0000 0000h. See [2.12.1 \[Machine Check Architecture\]](#).

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.



**MSR0000\_017B Global Machine Check Exception Reporting Control Register (MCG\_CTL)**

Reset: 0000 0000 0000 0000h. This register enables the various machine check register banks. See 2.12.1 [Machine Check Architecture]. When a machine check register bank is disabled, errors for that bank are not logged and actions controlled through the MCA are not taken. It is expected that this register is programmed to the same value in all nodes.

Bits	Description
63:6	<b>UnusedMcgCtl.</b> Read-only.
5	<b>FRE: fixed issue reorder buffer register bank enable.</b> Read-write. 1=The fixed-issue reorder buffer machine check register bank is enabled.
4	<b>NBE: Northbridge register bank enable.</b> Read-write. 1=The Northbridge machine check register bank is enabled.
3	<b>LSE: load-store register bank enable.</b> Read-write. 1=The load/store machine check register bank is enabled.
2	<b>BUE: bus unit register bank enable.</b> Read-write. 1=The bus unit machine check register bank is enabled.
1	<b>ICE: instruction cache register bank enable.</b> Read-write. 1=The instruction cache machine check register bank is enabled.
0	<b>DCE: data cache register bank enable.</b> Read-write. 1=The data cache machine check register bank is enabled.

**MSR0000\_01D9 Debug Control Register (DBG\_CTL\_MSR)**

Reset: 0000 0000 0000 0000h.

Bits	Description
63:7	Reserved.
6	MBZ.
5:2	<b>PB: performance monitor pin control.</b> Read-write. This field does not control any hardware.
1	<b>BTF.</b> Read-write. 1=Enable branch single step.
0	<b>LBR.</b> Read-write. 1=Enable last branch record.

**MSR0000\_01DB Last Branch From IP Register (BR\_FROM)**

Bits	Description
63:0	<b>LastBranchFromIP.</b> Read-only. Loaded with the segment offset of the branch instruction.

**MSR0000\_01DC Last Branch To IP Register (BR\_TO)**

Bits	Description
63:0	<b>LastBranchToIP.</b> Read-only. Holds the target RIP of the last branch that occurred before an exception or interrupt.

**MSR0000\_01DD Last Exception From IP Register**

Bits	Description
63:0	<b>LastIntFromIP.</b> Read-only. Holds the source RIP of the last branch that occurred before the exception or interrupt.

**MSR0000\_01DE Last Exception To IP Register**

Bits	Description
63:0	<b>LastIntToIP.</b> Read-only. Holds the target RIP of the last branch that occurred before the exception or interrupt.

**MSR0000\_02[0F:00] Variable-Size MTRRs (MTRRphysBasen and MTRRphysMaskn)**

Reset: xxxx xxxx xxxx xxxh. Each MTRR ([The Variable-Size MTRRs (MTRRphysBasen and MTRRphysMaskn)] MSR0000\_02[0F:00], [The Fixed-Size MTRRs (MTRRfixn)] MSR0000\_02[6F:68, 59, 58, 50], or [The MTRR Default Memory Type Register (MTRRdefType)] MSR0000\_02FF) specifies a physical address range and a corresponding memory type (MemType) associated with that range. Each 8-bit MemType field may include the following sub-fields:

- Bits[7:5]: reserved.
- Bit[4]: RdDram. 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. See 2.9.3 [Access Type Determination]. This bit can be enabled for fixed MTRR ranges only (see MSRC001\_0010[MtrrFixDramEn, MtrrFixDramModEn]); not variable-size MTRRs.
- Bit[3]: WrDram. 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. See 2.9.3 [Access Type Determination]. This bit can be enabled for fixed MTRR ranges only (see [The System Configuration Register (SYS\_CFG)] MSRC001\_0010); not variable-size MTRRs.
- Bits[2:0]: Memory type. The encodings for these are:
 

0h = UC or uncacheable.	5h = WP or write protect.
1h = WC or write combining.	6h = WB or write back.
4h = WT or write through.	All other values are reserved.

Setting MemType to an unsupported value results in a #GP(0).

The variable-size MTRRs come in pairs of base and mask registers (MSR0000\_0200 and MSR0000\_0201 are the first pair, etc.). Variables MTRRs are enabled through [The MTRR Default Memory Type Register (MTRRdefType)] MSR0000\_02FF[MtrrDefTypeEn]. A CPU access--with address CPUAddr--is determined to be within the address range of a variable-size MTRR if the following equation is true:

$$\text{CPUAddr}[47:12] \& \text{PhyMask}[47:12] == \text{PhyBase}[47:12] \& \text{PhyMask}[47:12].$$

For example, if the variable MTRR spans 256K bytes and starts at the 1M byte address. The PhyBase would be set to 00\_0010\_0000h and the PhyMask to FF\_FFFC\_0000h (with zeros filling in for bits[11:0]). This results in a range from 00\_0010\_0000h to 00\_0013\_FFFFh.

## MSR0000\_020[E, C, A, 8, 6, 4, 2, 0] (MTRRphysBasen)

Bits	Description
63:48	MBZ.
47:12	<b>PhyBase: base address.</b> Read-write.
11:8	MBZ.
7:0	<b>MemType: memory type.</b> Read-write.

## MSR0000\_020[F, D, B, 9, 7, 5, 3, 1] (MTRRphysMaskn)

Bits	Description
63:48	MBZ.
47:12	<b>PhyMask: address mask.</b> Read-write.
11	<b>Valid.</b> Read-write. 1=The variable-size MTRR pair is enabled.
10:0	MBZ.

**MSR0000\_02[6F:68, 59, 58, 50] Fixed-Size MTRRs (MTRRfixn)**

Reset: xxxx xxxx xxxx xxxh. See [MSR0000\\_02\[0F:00\]](#) for general MTRR information. Fixed MTRRs are enabled through [MSR0000\\_02FF](#)[MtrrDefTypeFixEn and MtrrDefTypeEn].

## MSR0000\_0250 (MTRRfix64K\_00000)

Bits	Description
63:56	<b>MemType: memory type.</b> Read-write. Address range from 7_0000 to 7_FFFF.
55:48	<b>MemType: memory type.</b> Read-write. Address range from 6_0000 to 6_FFFF.
47:40	<b>MemType: memory type.</b> Read-write. Address range from 5_0000 to 5_FFFF.
39:32	<b>MemType: memory type.</b> Read-write. Address range from 4_0000 to 4_FFFF.
31:24	<b>MemType: memory type.</b> Read-write. Address range from 3_0000 to 3_FFFF.
23:16	<b>MemType: memory type.</b> Read-write. Address range from 2_0000 to 2_FFFF.
15:8	<b>MemType: memory type.</b> Read-write. Address range from 1_0000 to 1_FFFF.
7:0	<b>MemType: memory type.</b> Read-write. Address range from 0_0000 to 0_FFFF.

## MSR0000\_0258 (MTRRfix16K\_80000) and MSR0000\_0259 (MTRRfix16K\_A0000)

The ranges specified below are described as offsets from the base address.

- The base address for MSR0000\_0258 = 8\_0000h.
- The base address for MSR0000\_0259 = A\_0000h.

Bits	Description
63:56	<b>MemType: memory type.</b> Read-write. Address range from 1_C000 to 1_FFFF (plus the base).
55:48	<b>MemType: memory type.</b> Read-write. Address range from 1_8000 to 1_BFFF (plus the base).
47:40	<b>MemType: memory type.</b> Read-write. Address range from 1_4000 to 1_7FFF (plus the base).
39:32	<b>MemType: memory type.</b> Read-write. Address range from 1_0000 to 1_3FFF (plus the base).
31:24	<b>MemType: memory type.</b> Read-write. Address range from 0_C000 to 0_FFFF (plus the base).

23:16	<b>MemType: memory type.</b> Read-write. Address range from 0_8000 to 0_BFFF (plus the base).
15:8	<b>MemType: memory type.</b> Read-write. Address range from 0_4000 to 0_7FFF (plus the base).
7:0	<b>MemType: memory type.</b> Read-write. Address range from 0_0000 to 0_3FFF (plus the base).

MSR0000\_02[6F:68] (MTRRfix4K\_XXXXX)

The ranges specified below are described as offsets from the base address.

- The base address for MSR0000\_0268 = C\_0000h.
- The base address for MSR0000\_0269 = C\_8000h.
- The base address for MSR0000\_026A = D\_0000h.
- The base address for MSR0000\_026B = D\_8000h.
- The base address for MSR0000\_026C = E\_0000h.
- The base address for MSR0000\_026D = E\_8000h.
- The base address for MSR0000\_026E = F\_0000h.
- The base address for MSR0000\_026F = F\_8000h.

Bits	Description
63:56	<b>MemType: memory type.</b> Read-write. Address range from 0_7000 to 0_7FFF (plus the base).
55:48	<b>MemType: memory type.</b> Read-write. Address range from 0_6000 to 0_6FFF (plus the base).
47:40	<b>MemType: memory type.</b> Read-write. Address range from 0_5000 to 0_5FFF (plus the base).
39:32	<b>MemType: memory type.</b> Read-write. Address range from 0_4000 to 0_4FFF (plus the base).
31:24	<b>MemType: memory type.</b> Read-write. Address range from 0_3000 to 0_3FFF (plus the base).
23:16	<b>MemType: memory type.</b> Read-write. Address range from 0_2000 to 0_2FFF (plus the base).
15:8	<b>MemType: memory type.</b> Read-write. Address range from 0_1000 to 0_1FFF (plus the base).
7:0	<b>MemType: memory type.</b> Read-write. Address range from 0_0000 to 0_0FFF (plus the base).

### MSR0000\_0277 Page Attribute Table Register (PAT)

Reset: 0007 0406 0007 0406h. This register specifies the memory type based on the PAT, PCD, and PWT bits in the virtual address page tables. The encodings for PA[7:0] is:

- |                                      |   |
|--------------------------------------|---|
| 0h = UC or uncacheable.              | 5h = WP or write protect.                             |
| 1h = WC or write combining.          | 6h = WB or write back.                                |
| 4h = WT or write through.            | 7h = UC- or uncacheable (overridden by MTRR WC state) |
| All other values result in a #GP(0). |   |

Bits	Description
63:59	MBZ.
58:56	<b>PA7 MemType.</b> Read-write. Default UC. MemType for {PAT, PCD, PWT} = 7h.
55:51	MBZ.
50:48	<b>PA6 MemType.</b> Read-write. Default UC-. MemType for {PAT, PCD, PWT} = 6h.
47:43	MBZ.
42:40	<b>PA5 MemType.</b> Read-write. Default WT. MemType for {PAT, PCD, PWT} = 5h.
39:35	MBZ.
34:32	<b>PA4 MemType.</b> Read-write. Default WB. MemType for {PAT, PCD, PWT} = 4h.
31:27	MBZ.

26:24	<b>PA3 MemType.</b> Read-write. Default UC. MemType for {PAT, PCD, PWT} = 3h.
23:19	MBZ.
18:16	<b>PA2 MemType.</b> Read-write. Default UC-. MemType for {PAT, PCD, PWT} = 2h.
15:11	MBZ.
10:8	<b>PA1 MemType.</b> Read-write. Default WT. MemType for {PAT, PCD, PWT} = 1h.
7:3	MBZ.
2:0	<b>PA0 MemType.</b> Read-write. Default WB. MemType for {PAT, PCD, PWT} = 0h.

### MSR0000\_02FF MTRR Default Memory Type Register (MTRRdefType)

Reset: 0000 0000 0000 0000h. See [MSR0000\\_02\[0F:00\]](#) for general MTRR information.

Bits	Description
63:12	MBZ.
11	<b>MtrrDefTypeEn: variable and fixed MTRR enable.</b> Read-write. 1=[ <a href="#">The Variable-Size MTRRs (MTRRphysBasen and MTRRphysMaskn)</a> ] <a href="#">MSR0000_02[0F:00]</a> , and [ <a href="#">The Fixed-Size MTRRs (MTRRfixn)</a> ] <a href="#">MSR0000_02[6F:68, 59, 58, 50]</a> , are enabled. 0=Fixed and variable MTRRs are not enabled.
10	<b>MtrrDefTypeFixEn: fixed MTRR enable.</b> Read-write. 1=[ <a href="#">The Fixed-Size MTRRs (MTRRfixn)</a> ] <a href="#">MSR0000_02[6F:68, 59, 58, 50]</a> are enabled. This field is ignored (and the fixed MTRRs are not enabled) if <a href="#">MSR0000_02FF[MtrrDefTypeEn]</a> =0.
9:8	MBZ.
7:0	<b>MemType: memory type.</b> Read-write. Specifies the memory type for space not mapped to enabled [ <a href="#">The Variable-Size MTRRs (MTRRphysBasen and MTRRphysMaskn)</a> ] <a href="#">MSR0000_02[0F:00]</a> , or enabled [ <a href="#">The Fixed-Size MTRRs (MTRRfixn)</a> ] <a href="#">MSR0000_02[6F:68, 59, 58, 50]</a> .

### MSR0000\_0400 DC Machine Check Control Register (MC0\_CTL)

Reset: 0000 0000 0000 0000h. All defined bits are read-write. BIOS should write FFFF FFFF FFFF FFFFh to initialize this register because some legacy operating systems do not initialize MC0\_CTL properly.

See [2.12.1 \[Machine Check Architecture\]](#). For all bits, 1=Enable the specified reporting mechanism.

Bits	Enable
63:8	<b>Unused.</b> Read-only.
7	<b>Unused.</b> Read-write.
6	<b>L2TP: L2 TLB parity errors.</b> Report data cache L2 TLB parity errors.
5	<b>L1TP: L1 TLB parity errors.</b> Report data cache L1 TLB parity errors.
4	<b>DSTP: snoop tag array parity errors.</b> Report data cache snoop tag array parity errors.
3	<b>DMTP: main tag array parity errors.</b> Report data cache main tag array parity errors.
2	<b>DECC: data array ECC errors.</b> Report data cache data array ECC errors. If not set, ECC errors in the cache are detected and logged, but not reported. If masked (see <a href="#">MSRC001_00[49:44]</a> ), ECC errors in the cache are undetected.

1	<b>ECCM: multi-bit ECC data errors.</b> Report multi-bit ECC data errors during data cache line fills from the internal L2 or the system. If masked (see <a href="#">MSRC001_00[49:44]</a> ), multi-bit ECC errors on line fills may be detected and logged as single-bit errors unless single-bit ECC data errors are also masked (ECCI). If masking all line fill data errors is desired, all ECC data error mask bits (ECCI and ECCM) must be set.
0	<b>ECCI: single-bit ECC data errors.</b> Report single-bit ECC data errors during data cache line fills from the internal L2 or the system. If masked (see <a href="#">MSRC001_00[49:44]</a> ), multi-bit ECC errors on line fills may also be masked. If masking all line fill data errors is desired, all ECC data error mask bits (ECCI and ECCM) must be set.

### MSR0000\_0401 DC Machine Check Status Register (MC0\_STATUS)

Cold reset: xxxx xxxx xxxx xxxh. See 2.12.1 [Machine Check Architecture]. Each of the MCI\_STATUS registers hold information identifying the last error logged in each bank. Software is normally only allowed to write 0's to these registers to clear the fields so subsequent errors may be logged. See [MSRC001\\_0015\[McStatusWrEn\]](#). The following field definitions apply to all MCI\_STATUS registers, except as noted.

Bits	Description
63	<b>Val: valid.</b> Read-write; set-by-hardware. 1=A valid error has been detected (whether it is enabled or not). This bit should be cleared to 0 by software after the register has been read.
62	<b>Over: error overflow.</b> Read-write; set-by-hardware. 1=An error was detected while the valid bit (Val) of this register was set; at least one error was not logged. The machine check mechanism handles the contents of MCI_STATUS during overflow as outlined in section 2.12.1.2.2 [Error Overflow].
61	<b>UC: error uncorrected.</b> Read-write; updated-by-hardware. 1=The error was not corrected by hardware.
60	<b>En: error enable.</b> Read-write; updated-by-hardware. 1=MCA error reporting is enabled for this error in MCI_CTL.
59	<b>MiscV: miscellaneous error register valid.</b> Read-only. 1=MCI_MISC contains valid information for this error. This bit is always 0, except in the case of [The NB Machine Check Misc (Thresholding) Register (MC4_MISC0)] <a href="#">MSR0000_0413</a> and [The FR Machine Check Miscellaneous Register (MC5_MISC)] <a href="#">MSR0000_0417</a> .
58	<b>AddrV: error address valid.</b> Read-write; updated-by-hardware. 1=The address saved in MCI_ADDR is the address where the error occurred.
57	<b>PCC: processor context corrupt.</b> Read-write; updated-by-hardware. 1=The state of the processor may have been corrupted by the error condition. Restart may not be reliable.
56:55	Reserved.
54:47	<b>Syndrome[7:0].</b> Read-write. <ul style="list-style-type: none"> <li>MC0_STATUS (DC): The lower eight syndrome bits when an ECC error is detected. See <a href="#">Table 79</a> for the mappings that show which bit errors result in which syndrome values.</li> <li>MC[3:1]_STATUS (LS, BU, IC): Reserved.</li> </ul>
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. <a href="#">MSR0000_0401[UC]</a> 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.

44:41	Reserved.
40	<b>Scrub: error detected on a scrub.</b> Read-write; updated-by-hardware. <ul style="list-style-type: none"> <li>• MC0_STATUS (DC), MC2_STATUS (BU): 1=The error was detected on a scrub.</li> <li>• MC1_STATUS (IC), MC3_STATUS (LS), MC5_STATUS (FR): Reserved.</li> </ul>
39:32	Reserved.
31:24	<b>Syndrome[15:8].</b> Read-write. <ul style="list-style-type: none"> <li>• MC0_STATUS (DC): The upper eight syndrome bits when an ECC error is detected. See <a href="#">Table 79</a> for the mappings that show which bit errors result in which syndrome values.</li> <li>• MC[3:1]_STATUS (LS, BU, IC): Reserved.</li> </ul>
23:20	Reserved.
19:16	<b>ErrorCodeExt: extended error code.</b> Read-write. See the appropriate error signature tables below: <ul style="list-style-type: none"> <li>• MC0_STATUS (DC): <a href="#">Table 107</a></li> <li>• MC1_STATUS (IC): <a href="#">Table 110</a></li> <li>• MC2_STATUS (BU): <a href="#">Table 113</a></li> <li>• MC3_STATUS (LS): <a href="#">Table 115</a></li> <li>• MC5_STATUS (FR): <a href="#">Table 116</a></li> </ul>
15:0	<b>ErrorCode: error code.</b> Read-write. See the appropriate error signature tables below: <ul style="list-style-type: none"> <li>• MC0_STATUS (DC): <a href="#">Table 107</a></li> <li>• MC1_STATUS (IC): <a href="#">Table 110</a></li> <li>• MC2_STATUS (BU): <a href="#">Table 113</a></li> <li>• MC3_STATUS (LS): <a href="#">Table 115</a></li> <li>• MC5_STATUS (FR): <a href="#">Table 116</a></li> </ul>

This register reports these DC errors:

**Table 106: DC error descriptions**

Error Type	Description	Enablers (MSR0000_0400 Control Bits)
L2 Cache Line Fill	An error occurred during an L1 line fill from the L2 cache.	ECC1, ECCM.
Data Load/ Store/ Victim/ Snoop	A data error occurred while accessing or managing data. The CECC and UECC fields of the status register indicate whether a correctable or uncorrectable ECC error was encountered. Note that not all CECC errors are correctable in all flows and processor configurations; the UC field of the status register indicates whether the processor was able to correct the error.	DECC
Data Scrub	An error was detected during a scrub of cache data.	DECC
Tag Snoop/ Victim	A tag error was encountered during snoop or victimization.	DSTP
Tag Load/Store	A tag error was encountered during load or store.	
L1 TLB	Parity error in L1 TLB.	L1TP
L1 TLB Multi-match	Hit multiple entries.	L1TP
L2 TLB	Parity error in L2 TLB.	L2TP
L2 TLB Multi-match	Hit multiple entries.	L2TP



**Table 107: DC error signatures**

Error Type	[19:16] Error-CodeExt	Error Code (see F3x48 for encoding)						[61] UC	[58] ADD-RV	[57] PCC	[54:47] Synd Valid	[46] CECC	[45] UECC	[40] Scrub
		Type	10:9 PP	8 T	7:4 RRRR	3:2 II/TT	1:0 LL							
L2 Cache Line Fill	0000	Mem	-	-	DRD	Data	L2	If multi-bit	1	1/0	Y	If single-bit	If multi-bit	0
Data Load/Store/Victim/Snoop	0000	Mem	-	-	DRD/DWR/Evict/Snoop	Data	L1	If not corrected	1/0	1/0	Y	If single-bit	If multi-bit	0
Data Scrub	0000	Mem	-	-	GEN	Data	L1	If multi-bit	1	0	Y	If single-bit	If multi-bit	1
Tag Snoop/Victim	0000	Mem	-	-	Snoop/Evict	Data	L1	1	1/0	1	N	0	0	0
Tag Load/Store	0000	Mem	-	-	DRD/DWR	Data	L1	1	1	1	N	0	0	0
L1 TLB	0000	TLB	-	-	-	Data	L1	1	1	1	N	0	0	0
L1 TLB Multimatch	0001	TLB	-	-	-	Data	L1	1	1	1	N	0	0	0
L2 TLB	0000	TLB	-	-	-	Data	L2	1	1	1	N	0	0	0
L2 TLB Multimatch	0001	TLB	-	-	-	Data	L2	1	1	1	N	0	0	0

**MSR0000\_0402 DC Machine Check Address Register (MC0\_ADDR)**

Cold reset: xxxx xxxx xxxx xxxh. See 2.12.1 [Machine Check Architecture]. Each of the MCi\_ADDR registers are written to by hardware and read-write accessible by software. MCi\_ADDR registers contains valid data if indicated by MCi\_STATUS[AddrV]. Table 108 defines the address register as a function of error type.

**Table 108: DC error data; address register**

Error Type	Memory Transaction Type (RRRR; Table 89)	Address Register Bits	Description
L2 Cache Line Fill	DRD	47:6	Physical address
Data Load/Store/Victim/Snoop	DRD	47:4 <sup>1</sup>	Physical address
	DWR		
	Evict/Snoop	11:6	Physical address
Data Scrub	GEN	11:4	Physical address

- For Data Store (DWR), address bits shown are present only if error was reported (MSR0000\_0401[UC] is set and MSR0000\_0400[DECC] is enabled and not masked). If not reported, then valid address register bits are the linear address in 14:4.
- The entire address from the TLB may be stored, but that address may only be incidentally related to the tag error; only the indicated bits are valid for this type of error.

**Table 108: DC error data; address register**

Error Type	Memory Transaction Type (RRRR; Table 89)	Address Register Bits	Description
Tag Snoop/ Victim	Snoop	11:6	Physical address
	Evict		
Tag Load/ Store	DRD	11:6 <sup>2</sup>	Physical address
	DWR		
L1 TLB	-	47:12	Linear address
L1 TLB Multi-match			
L2 TLB			
L2 TLB Multi-match			
<p>1. For Data Store (DWR), address bits shown are present only if error was reported (<b>MSR0000_0401</b>[UC] is set and <b>MSR0000_0400</b>[DECC] is enabled and not masked). If not reported, then valid address register bits are the linear address in 14:4.</p> <p>2. The entire address from the TLB may be stored, but that address may only be incidentally related to the tag error; only the indicated bits are valid for this type of error.</p>			

**MSR0000\_0403 DC Machine Check Miscellaneous Register (MC0\_MISC)**

This register is read-only, reset: 0000 0000 0000 0000h.

**MSR0000\_0404 IC Machine Check Control Register (MC1\_CTL)**

Reset: 0000 0000 0000 0000h. All defined bits are read-write.

See 2.12.1 [Machine Check Architecture]. For all bits, 1=Enable the specified reporting mechanism.

Bits	Enable
63:32	<b>Unused.</b> Read-only.
31:10	<b>Unused.</b> Read-write.
9	<b>RDDE: read data errors.</b> Report system read data errors for an instruction cache fetch if [The BU Machine Check Control Register (MC2_CTL)] <b>MSR0000_0408</b> [SRDE_ALL] = 1.
8	<b>Unused.</b> Read-only.
7	<b>Unused.</b> Read-write.
6	<b>L2TP: L2 TLB parity errors.</b> Report instruction cache L2 TLB parity errors.
5	<b>L1TP: L1 TLB parity errors.</b> Report instruction cache L1 TLB parity errors.
4	<b>ISTP: snoop tag array parity errors.</b> Report instruction cache snoop tag array parity errors.
3	<b>IMTP: main tag array parity errors.</b> Report instruction cache main tag array parity errors.
2	<b>IDP: data array parity errors.</b> Report instruction cache data array parity errors.
1	<b>ECCM: multi-bit ECC data errors.</b> Report multi-bit ECC data errors during instruction cache line fills or TLB reloads from the internal L2 or the system.

Bits	Enable
0	<b>ECCI: single-bit ECC data errors.</b> Report single-bit ECC data errors during instruction cache line fills or TLB reloads from the internal L2 or the system.

### MSR0000\_0405 IC Machine Check Status Register (MC1\_STATUS)

Cold reset: xxxx xxxx xxxx xxxh. See 2.12.1 [Machine Check Architecture]. See MSR0000\_0401 for the information about all of the MCI\_STATUS registers. See also MSRC001\_0015[McStatusWrEn]. This register reports these IC errors:

**Table 109: IC error descriptions**

Error Type	Description	Enablers (MSR0000_0404 Control Bits)
System Data Read Error	An error occurred during an attempted read of data from the NB. Possible reasons include master abort, target abort.	RDDE
L2 Cache Line Fill	An error occurred during a line fill from the L2 cache.	ECCM
IC Data Load (Parity)	A parity error occurred during load of data from the IC. This may be either a data error or a tag error. The data is discarded from the IC and can be refetched.	IDP, IMTP
Tag Snoop	A tag error was encountered during snoop or victimization.	ISTP
Copyback parity	A copyback parity error occurred.	IMTP
L1 TLB	Parity error in L1 TLB.	L1TP
L1 TLB Multi-match	Hit multiple entries.	L1TP
L2 TLB	Parity error in L2 TLB.	L2TP
L2 TLB Multi-match	Hit multiple entries.	L2TP

**Table 110: IC error signatures**

Error Type	[19:16] Error-CodeExt	Error Code (see F3x48 for encoding)							[61] UC	[58] ADD-RV	[57] PCC	[54:47] Synd Valid	[46] CECC	[45] UECC	[40] Scrub
		Type	10:9 PP	8 T	7:4 RRRR	3:2 II/TT	1:0 LL								
System Data Read Error	0000	BUS	SRC	0	IRD	MEM	LG	1	0	0	N	0	0	0	
L2 Cache Line Fill	0000	Memory	-	-	IRD	Instr	L2	0 <sup>1</sup>	1	0	N	0 <sup>2</sup>	1	0	
IC Data Load (Parity)	0000	Memory	-	-	IRD	Instr	L1	0	1	0	N	0	0	0	
Tag Snoop	0000	Memory	-	-	Snoop	Instr	L1	1	1	1	N	0	0	0	

1. Line refetched from memory. (Automatically purged from L2 during fill.)  
 2. Single bit errors are detected as parity errors.

**Table 110: IC error signatures**

Error Type	[19:16] Error-CodeExt	Error Code (see F3x48 for encoding)							[61] UC	[58] ADD-RV	[57] PCC	[54:47] Synd Valid	[46] CECC	[45] UECC	[40] Scrub
		Type	10:9 PP	8 T	7:4 RRRR	3:2 II/TT	1:0 LL								
Copyback parity	0000	Mem-ory	-	-	Evict	Instr	L1	0	0	0	N	0	0	0	
L1 TLB	0000	TLB	-	-	-	Instr	L1	0	1	0	N	0	0	0	
L1 TLB Multimatch	0001	TLB	-	-	-	Instr	L1	0	1	0	N	0	0	0	
L2 TLB	0000	TLB	-	-	-	Instr	L2	0	1	0	N	0	0	0	
L2 TLB Multimatch	0001	TLB	-	-	-	Instr	L2	0	1	0	N	0	0	0	

- Line refetched from memory. (Automatically purged from L2 during fill.)
- Single bit errors are detected as parity errors.

**MSR0000\_0406 IC Machine Check Address Register (MC1\_ADDR)**

Cold reset: xxxx xxxx xxxx xxxh. See 2.12.1 [Machine Check Architecture]. Each of the MCi\_ADDR registers are written to by hardware and read-write accessible by software. MCi\_ADDR registers contains valid data if indicated by MCi\_STATUS[AddrV]. Table 111 defines the address register as a function of error type.

**Table 111: IC error data; address register**

Error Type	Address Register Bits	Description
L2 Cache Line Fill	47:6	Physical address
IC Data Load	47:4	Linear address
Tag Snoop	47:6	Physical address
L1 TLB	47:12 for 4-Kbyte page	Linear address
L1 TLB Multi-match	47:20 for 2-Mbyte page	
L2 TLB	47:12 for 4-Kbyte page	Linear address
L2 TLB Multi-match		

**MSR0000\_0407 IC Machine Check Miscellaneous Register (MC1\_MISC)**

This register is read-only, reset: 0000 0000 0000 0000h.

**MSR0000\_0408 BU Machine Check Control Register (MC2\_CTL)**

Reset: 0000 0000 0000 0000h. All defined bits are read-write.

See 2.12.1 [Machine Check Architecture]. For all bits, 1=Enable the specified reporting mechanism.

Bits	Enable
63:12	<b>Unused.</b> Read-only
11	<b>PDC_PAR: Pdc/GTLB parity errors.</b> Report Page Descriptor Cache parity or Guest TLB table walk parity errors.

Bits	Enable
10	<b>VB_PAR: write/victim data buffer parity error.</b> Report write buffer or victim buffer data parity errors.
9	<b>Unused.</b> Read-write.
8	<b>L2D_UECC: L2 data uncorrectable ECC error.</b> Report L2 data array uncorrectable ECC errors.
7	<b>L2D_CECC: L2 data correctable ECC error.</b> Report L2 data array correctable ECC errors.
6	<b>L2D_PAR: L2 data parity errors.</b> Report correctable and uncorrectable L2 data array parity errors.
5	<b>L2T_UECC: L2 tag uncorrectable ECC error.</b> Report L2 tag array uncorrectable ECC errors.
4	<b>L2T_CECC: L2 tag correctable ECC error.</b> Report L2 tag array correctable ECC errors.
3	<b>L2T_PAR: L2 tag parity errors.</b> Report L2 tag array correctable and uncorrectable parity errors.
2	<b>SRDE_ALL: all system read data.</b> Report system read data errors for any operation including a DC/IC fetch, TLB reload or hardware prefetch.
1	<b>SRDE_TLB: system read data TLB reload.</b> Report system read data errors for a TLB reload.
0	<b>SRDE_HP: system read data hardware prefetch.</b> Report system read data errors for a hardware prefetch.

#### MSR0000\_0409 BU Machine Check Status Register (MC2\_STATUS)

Cold reset: xxxx xxxx xxxx xxxh. See 2.12.1 [Machine Check Architecture]. See MSR0000\_0401 for the information about all of the MCI\_STATUS registers. See also MSRC001\_0015[McStatusWrEn]. This register reports these BU errors:

**Table 112: BU error descriptions**

Error Type	Description	Enablers (MSR0000_0408 Control Bits)
System Data Read Error	An error occurred during an attempted read of data from the NB. Possible reasons include master abort and target abort.	SRDE_ALL, SRDE_HP, SRDE_TLB
L2 Cache Data	A parity or ECC error occurred during a data access from the L2 cache.	L2D_CECC, L2D_UECC, L2D_PAR
Data Buffer	An error occurred in the write or victim data buffers.	VB_PAR
Data Copyback	An error occurred on a data copyback.	L2D_CECC, L2D_UECC, L2D_PAR
Tag	An error occurred in the L2 cache tags.	L2T_PAR, L2T_CECC, L2T_UECC
PDC/GTLB Parity	A parity error occurred in a PDC or GTLB.	PDC_PAR

**Table 113: BU error signatures**

Error Type	Access Type	[19:16] Error-CodeExt	Error Code (see F3x48 for encoding)						[61] UC	[58] ADD-RV	[57] PCC	[54:47] Synd Valid	[46] CECC	[45] UECC	[40] Scrub
			Type	10:9 PP	8 T	7:4 RRRR	3:2 II/TT	1:0 LL							
System Data Read Error	TLB	0000	BUS	SRC	0	RD	MEM/IO	LG	1	1	0	N	0	0	0
	HW Prefetch	0000	BUS	SRC	0	Prefetch	MEM/IO	LG	1	0	0	N	0	0	0
L2 Cache Data	TLB	0000	Mem	-	-	RD	Gen	L2	1/0	1	0	N	1/0	1/0	0
	Scrub	0000	Mem	-	-	GEN	Gen	L2	1/0	1	0	N	1/0	1/0	1
Data buffer	Victim	0011	Mem	-	-	Snoop/Evict	Gen	LG	1/0	1/0	If UC	N	0	0	0
	Write	0001	Mem	-	-	WR	Gen	LG	1/0	1/0	If UC	N	0	0	0
Data Copy-back	Snoop/Evict	0000	Mem	-	-	Snoop/Evict	Gen	L2	1/0	1	If UC	N	1/0	1/0	0
Tag	Instr Fetch	0010	Mem	-	-	IRD	Instr	L2	1/0	1	If UC	N	1/0	1/0	0
	Data Fetch	0010	Mem	-	-	DRD	Data	L2	1/0	1	If UC	N	1/0	1/0	0
	TLB/Snoop/Evict	0010	Mem	-	-	RD/Snoop/Evict	Gen	L2	1/0	1	If UC	N	1/0	1/0	0
	Scrub	0010	Mem	-	-	GEN	Gen	L2	1/0	1	0	N	1/0	1/0	1
PDC and GTLB Parity Error	Instr Fetch	0000	TLB	-	-	-	Instr	L1	1/0	1/0	1/0	N	0	0	0
	Data Fetch	0000	TLB	-	-	-	Data	L1	1/0	1/0	1/0	N	0	0	0

**MSR0000\_040A BU Machine Check Address Register (MC2\_ADDR)**

Cold reset: xxxx xxxx xxxx xxxh. See 2.12.1 [Machine Check Architecture]. Each of the MCi\_ADDR registers are written to by hardware and read-write accessible by software. MCi\_ADDR registers contains valid data if indicated by MCi\_STATUS[AddrV]. Table 114 defines the address register as a function of error type

**Table 114: BU error data; address register**

Error Type	Address Register Bits	Description
System Data Read Error	47:6	Physical address
L2 Cache Data		
Data buffers		
Data copyback		
Tag	3:0	Encoded cache way
	15:6 for 1-Mbyte L2	Physical address
	14:6 for 512-Kbyte L2	
	13:6 for 256-Kbyte L2	
12:6 for 128-Kbyte L2		
PDC/Guest TLB parity error	47:2	TLB reloader access or fetch address

**MSR0000\_040B BU Machine Check Miscellaneous Register (MC2\_MISC)**

This register is read-only, reset: 0000 0000 0000 0000h.

**MSR0000\_040C LS Machine Check Control Register (MC3\_CTL)**

Reset: 0000 0000 0000 0000h. All defined bits are read-write.

See 2.12.1 [Machine Check Architecture]. For all bits, 1=Enable the specified reporting mechanism.

Bits	Enable
63:2	<b>Unused.</b> Read-only.
1	<b>SRDE_S: read data errors on store.</b> Report system read data errors on a store if [The BU Machine Check Control Register (MC2_CTL)] MSR0000_0408[SRDE_ALL] = 1.
0	<b>SRDE_L: read data errors on load.</b> Report system read data errors on a load if [The BU Machine Check Control Register (MC2_CTL)] MSR0000_0408[SRDE_ALL] = 1.

**MSR0000\_040D LS Machine Check Status Register (MC3\_STATUS)**

Cold reset: xxxx xxxx xxxx xxxh. See 2.12.1 [Machine Check Architecture]. See MSR0000\_0401 for the information about all of the MCi\_STATUS registers. See also MSRC001\_0015[McStatusWrEn]. This register reports these LS errors:

**Table 115: LS error signatures**

Error Type	[19:16] Error-CodeExt	Error Code (see F3x48 for encoding)						[61] UC	[58] ADD-RV	[57] PCC	[54:47] Synd Valid	[46] CECC	[45] UECC	[40] Scrub
		Type	10:9 PP	8 T	7:4 RRRR	3:2 II/TT	1:0 LL							
Read Data on Store	0000	BUS	SRC	0	DWR	MEM	LG	1	1/0	1/0	N	0	0	0
Read Data on Load	0000	BUS	SRC	0	DRD	MEM/IO	LG	1	1/0	1/0	N	0	0	0

**MSR0000\_040E LS Machine Check Address Register (MC3\_ADDR)**

Cold reset: xxxx xxxx xxxx xxxh. See 2.12.1 [Machine Check Architecture]. Each of the MCi\_ADDR registers are written to by hardware and read-write accessible by software. MCi\_ADDR registers contains valid data if indicated by MCi\_STATUS[AddrV]. The only type of error recorded by the LS machine check mechanism is a “system address out of range” or read data error for which MC3\_ADDR[47:0] store the physical address.

**MSR0000\_040F LS Machine Check Miscellaneous Register (MC3\_MISC)**

This register is read-only, reset: 0000 0000 0000 0000h.

**MSR0000\_0410 NB Machine Check Control Register (MC4\_CTL)**

See 2.12.1 [Machine Check Architecture]. MSR0000\_0410[31:0] is a copy of [The MCA NB Control Register] F3x40. Only one of these registers exists in multi-core devices; see 3.1.1 [Northbridge MSRs In Multi-Core Products].



Bits	Enable
63:32	<b>Unused.</b> Read-only.
31:0	See [The MCA NB Control Register] F3x40.

### **MSR0000\_0411 NB Machine Check Status Register (MC4\_STATUS)**

See 2.12.1 [Machine Check Architecture]. MSR0000\_0411[31:0] is a copy of [The MCA NB Status Low Register] F3x48. MSR0000\_0411[63:32] is a copy of [The MCA NB Status High Register] F3x4C. Only one of these registers exists in multi-core devices; see 3.1.1 [Northbridge MSRs In Multi-Core Products].

Bits	Enable
63:32	See F3x4C.
31:0	See F3x48.

### **MSR0000\_0412 NB Machine Check Address Register (MC4\_ADDR)**

See 2.12.1 [Machine Check Architecture]. MSR0000\_0412[31:0] is a copy of [The MCA NB Address Low Register] F3x50 and MSR0000\_0412[63:32] is a copy of [The MCA NB Address High Register] F3x54. Only one of these registers exists in multi-core devices; see 3.1.1 [Northbridge MSRs In Multi-Core Products].

Bits	Enable
63:32	See F3x54.
31:0	See F3x50.

### **MSR0000\_0413 NB Machine Check Misc (Thresholding) Register (MC4\_MISC0)**

MSR0000\_0413 is the first of the NB machine check miscellaneous registers. MSR0000\_0413[31:0] is expanded below; MSR0000\_0413[63:32] is a duplicate access method to NB register F3x160[31:0] (see F3x1[78, 70, 68, 60]).

To see the remaining NB machine check miscellaneous registers, refer to [The Machine Check Misc 4 (Thresholding) Registers 1 to 3 (MC4\_MISC[3:1])] MSRC000\_04[0A:08]. For general information on error thresholding, see 2.12.1.6 [Error Thresholding].

Only one of these registers exists in multi-core devices; see 3.1.1 [Northbridge MSRs In Multi-Core Products].

Bits	Enable
63:32	See F3x1[78, 70, 68, 60].
31:24	<b>BlkPtr: Block pointer for additional MISC registers.</b> Read-only. Valid only when Valid field set. When non-zero, used to calculate a pointer to the extended MISC MSR block (see 2.12.1.1 [Machine Check Registers]) as follows: MC4_MISC1 = (MC4_MISC[BlkPtr] shifted left 3 bits) + C000_0400h.
23:0	Reserved.

### **MSR0000\_0414 FR Machine Check Control Register (MC5\_CTL)**

Reset: 0000 0000 0000 0000h. All defined bits are read-write.

See 2.12.1 [Machine Check Architecture]. For all bits, 1=Enable the specified reporting mechanism.

Bits	Enable
63:1	Reserved.
0	<b>CPUWDT: CPU watchdog timer.</b> The core WDT expiration (see [The CPU Watchdog Timer Register (CpuWdTmrCfg)] MSRC001_0074).

### MSR0000\_0415 FR Machine Check Status Register (MC5\_STATUS)

Cold reset: xxxx xxxx xxxx xxxh. See 2.12.1 [Machine Check Architecture]. See MSR0000\_0401 for the information about all of the MCi\_STATUS registers. See also MSRC001\_0015[McStatusWrEn]. This register reports these FR errors:

**Table 116: FR error signatures**

Error Type	[19:16] Error-CodeExt	Error Code (see F3x48 for encoding)						[61] UC	[58] ADD-RV	[57] PCC	[54:47] Synd Valid	[46] CECC	[45] UECC	[40] Scrub
		Type	10:9 PP	8 T	7:4 RRRR	3:2 II/TT	1:0 LL							
CPU watchdog timer expire	----	Bus	Gen	1	GEN	Gen	LG	1	1	1	No	0	0	0

### MSR0000\_0416 FR Machine Check Address Register (MC5\_ADDR)

Cold reset: xxxx xxxx xxxx xxxh. See 2.12.1 [Machine Check Architecture]. Each of the MCi\_ADDR registers are written to by hardware and read-write accessible by software. MCi\_ADDR registers contains valid data if indicated by MCi\_STATUS[AddrV]. The only type of error recorded by the FR machine check mechanism results in a load of the logical address of the next instruction after the last instruction retired in MC5\_ADDR[47:0].

### MSR0000\_0417 FR Machine Check Miscellaneous Register (MC5\_MISC)

Reset: xxxx xxxx xxxx xxxh. This register records unspecified, implementation-specific status bits when an FR machine check error is logged.

## 3.11 MSRs - MSRC000\_0xxx

### MSRC000\_0080 Extended Feature Enable Register (EFER)

Reset: 0000 0000 0000 0000h.

SKINIT Execution: 0000 0000 0000 0000h.

Bits	Description
63:15	MBZ.
14	<b>FFXSE: fast FXSAVE/FRSTOR enable.</b> Read-write. 1=Enables the fast FXSAVE/FRSTOR mechanism. A 64-bit operating system uses <code>CPUID Fn[8000_0001, 0000_0001]_EDX[24]</code> to determine the presence of this feature before enabling it. This bit is set once by the operating system and its value is not changed afterwards.
13	<b>LMSLE: long mode segment limit enable.</b> Read-write. 1=Enables the long mode segment limit check mechanism.
12	<b>SVME: secure virtual machine (SVM) enable.</b> Read-write. 1=SVM features are enabled.

11	<b>NXE: no-execute page enable.</b> Read-write. 1=The no-execute page protection feature is enabled.
10	<b>LMA: long mode active.</b> Read-only. 1=Indicates that long mode is active.
9	MBZ.
8	<b>LME: long mode enable.</b> Read-write. 1=Long mode is enabled.
7:1	RAZ.
0	<b>SYSCALL: system call extension enable.</b> Read-write. 1=SYSCALL and SYSRET instructions are enabled. This adds the SYSCALL and SYSRET instructions which can be used in flat addressed operating systems as low latency system calls and returns.

### MSRC000\_0081 SYSCALL Target Address Register (STAR)

Reset: 0000 0000 0000 0000h. This register holds the target address used by the SYSCALL instruction and the code and stack segment selector bases used by the SYSCALL and SYSRET instructions.

Bits	Description
63:48	<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 Register (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 Register (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 Register (SYSCALL\_FLAG\_MASK)

Reset: 0000 0000 0000 0000h.

Bits	Description
63:32	RAZ.
31:0	<b>MASK: SYSCALL flag mask.</b> Read-write. This register holds the EFLAGS mask used by the SYSCALL instruction. 1=Clear the corresponding EFLAGS bit when executing the SYSCALL instruction.

**MSRC000\_0100 FS Base Register (FS\_BASE)**

Reset: 0000 0000 0000 0000h.

Bits	Description
63:0	<b>FS_BASE: expanded FS segment base.</b> Read-write. This register provides access to the expanded 64-bit FS segment base. The address stored in this register must be in canonical form (if not canonical, a #GP fault fill occurs).

**MSRC000\_0101 GS Base Register (GS\_BASE)**

Reset: 0000 0000 0000 0000h.

Bits	Description
63:0	<b>GS_BASE: expanded GS segment base.</b> Read-write. This register provides access to the expanded 64-bit GS segment base. The address stored in this register must be in canonical form (if not canonical, a #GP fault fill occurs).

**MSRC000\_0102 Kernel GS Base Register (KernelGSbase)**

Reset: 0000 0000 0000 0000h.

Bits	Description
63:0	<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 Register (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\_04[0A:08] Machine Check Misc 4 (Thresholding) Registers 1 to 3 (MC4\_MISC[3:1])**

MSRC000\_04[0A:08] are the block of extended NB machine check miscellaneous registers.

MSRC000\_04[0A:08][31:0] are mapped identically to MSR0000\_0413[31:0]; MSRC000\_04[0A:08][63:32] are duplicate access methods to the corresponding registers of F3x1[78, 70, 68, 60][31:0].

Only one of these register blocks exists in multi-core devices; see 3.1.1 [Northbridge MSRs In Multi-Core Products].

Bits	Description
63:32	See F3x1[78, 70, 68, 60][31:0].
31:0	See MSR0000_0413[31:0].

### MSRC000\_04[0F:0B] Reserved

Bits	Description
63:0	RAZ.

## 3.12 MSRs - MSRC001\_0xxx

### MSRC001\_00[03:00] Performance Event Select Register (PERF\_CTL[3:0])

Reset: xxxx xxxx xxxx xxxh. PERF\_CTL[3:0] are used to specify the events counted by the [The Performance Event Counter Registers (PERF\_CTR[3:0])] MSRC001\_00[07:04] and to control other aspects of their operation. Each performance counter supported has a corresponding event-select register that controls its operation. Section 3.14 [Performance Counter Events] shows the events and unit masks supported by the processor.

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 zero and the second event is a static one. To avoid this false edge detection, disable the counter when changing the event and then enable the counter with a second MSR write.

The performance counter registers can be used to track events in the Northbridge. Northbridge events include all memory controller events, crossbar events, and HyperTransport™ interface events as documented in 3.14.7, 3.14.8, and 3.14.9. Monitoring of Northbridge events should only be performed by one core. If a Northbridge event is selected using one of the Performance Event-Select registers in any core of a multi-core processor, then a Northbridge performance event cannot be selected in the same Performance Event Select register of any other core.

Care must be taken when measuring Northbridge or other non-processor-specific events under conditions where the processor may go into halt mode during the measurement period. For instance, one may wish to monitor DRAM traffic due to DMA activity from a disk or graphics adaptor. This entails running some event counter monitoring code on the processor, where such code accesses the counters at the beginning and end of the measurement period, or may even sample them periodically throughout the measurement period. Such code typically gives up the processor during each measurement interval. If there is nothing else for the OS to run on that particular processor at that time, it may halt the processor until it is needed. Under these circumstances, the clock for the counter logic may be stopped, hence the counters would not count the events of interest. To prevent this, simply run a low-priority background process that keeps the processor busy during the period of interest.

Bits	Description
63:42	Reserved.
41	<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. 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. 01h-03h 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. 04h-FFh 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.
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_CTR[3:0] register. The events are specified in section 3.14 [Performance Counter Events]. Some events are reserved; when a reserved event is selected, the results are undefined.

### **MSRC001\_00[07:04] Performance Event Counter Registers (PERF\_CTR[3:0])**

Reset: 0000 xxxx xxxx xxxh. The processor provides four 48-bit performance counters. Each counter can monitor a different event specified by [The Performance Event Select Register (PERF\_CTL[3:0])] MSRC001\_00[03:00]. The accuracy of the counters is not ensured.

Performance counters are used to count specific processor events, such as data-cache misses, or the duration of events, such as the number of clocks it takes to return data from memory after a cache miss. During event counting, the processor increments the counter when it detects an occurrence of the event. During duration measurement, the processor counts the number of processor clocks it takes to complete an event. Each performance counter can be used to count one event, or measure the duration of one event at a time.

In addition to the RDMSR instruction, the PERF\_CTR[3:0] registers can be read using a special read performance-monitoring counter instruction, RDPMC. The RDPMC instruction loads the contents of the PERF\_CTR[3:0] register specified by the ECX register, into the EDX register and the EAX register.

Writing the performance counters can be useful if there is an intention for software to count a specific number of events, and then trigger an interrupt when that count is reached. An interrupt can be triggered when a performance counter overflows. Software should use the WRMSR instruction to load the count as a two's-complement negative number into the performance counter. This causes the counter to overflow after counting the appropriate number of times.

The performance counters are not assured of producing identical measurements each time they are used to measure a particular instruction sequence, and they should not be used to take measurements of very small instruction sequences. The RDPMC instruction is not serializing, and it can be executed out-of-order with respect to other instructions around it. Even when bound by serializing instructions, the system environment at the time the instruction is executed can cause events to be counted before the counter value is loaded into EDX:EAX.

Bits	Description
63:48	RAZ.
47:0	<b>CTR: performance counter value.</b> Read-write. Returns the current value of the event counter.

#### MSRC001\_0010 System Configuration Register (SYS\_CFG)

Reset: 0000 0000 0002 0600h.

Bits	Description
63:23	Reserved.
22	<b>Tom2ForceMemTypeWB: top of memory 2 memory type write back.</b> Read-write. 1=The default memory type of memory between 4GB and TOM2 is write back instead of the memory type defined by [The MTRR Default Memory Type Register (MTRRdefType)] MSR0000_02FF[MemType]. For this bit to have any effect, MSR0000_02FF[MtrrDefTypeEn] must be 1. MTRRs and PAT can be used to override this memory type.
21	<b>MtrrTom2En: MTRR top of memory 2 enable.</b> Read-write. 0=[The Top Of Memory 2 Register (TOM2)] MSRC001_001D is disabled. 1=This register is enabled.
20	<b>MtrrVarDramEn: MTRR variable DRAM enable.</b> Read-write. 0=[The Top Of Memory Register (TOP_MEM)] MSRC001_001A and IORRs are disabled. 1=These registers are enabled. This bit should be set by BIOS.
19	<b>MtrrFixDramModEn: MTRR fixed RdDram and WrDram modification enable.</b> Read-write. 0=Reads from the RdDram and WrDram bits of [The Fixed-Size MTRRs (MTRRfixn)] MSR0000_02[6F:68, 59, 58, 50] return 00b and writes of those bits are ignored. 1=These bits are read-write accessible. 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. 1=Enables the RdDram and WrDram attributes in [The Fixed-Size MTRRs (MTRRfixn)] MSR0000_02[6F:68, 59, 58, 50]. This bit should be set by BIOS.
17	<b>SysUcLockEn: system lock command enable.</b> Read-write. 1=Transactions to the coherent fabric support the lock command. This is normally enabled in multi-core systems and disabled in single core systems.



16	<b>ChgToDirtyDis: change to dirty disable.</b> Read-write. 1=Disables Change-to-Dirty commands, evicts line from DC instead.
15:11	Reserved.
10	<b>SetDirtyEnO: clean-to-dirty command for O-&gt;M state transition enable.</b> Read-write. 1=Enables generating write probes when transitioning a cache line from Owned to Modified.
9	<b>SetDirtyEnS: shared-to-dirty command for S-&gt;M state transition enable.</b> Read-write. 1=Enables generating write probes when transitioning a cache line from Shared to Modified.
8	<b>SetDirtyEnE: shared-to-dirty command for E-&gt;M state transition enable.</b> Read-write. 1=Enables generating write probes when transitioning a cache line from Exclusive to Modified.
7:0	Reserved.

### MSRC001\_0015 Hardware Configuration Register (HWCR)

Reset: 0000 0000 0000 0010h.

Bits	Description
63:27	Reserved.
26	<b>EffFreqCntMwait: effective frequency counting during mwait.</b> Revision D and earlier: Reserved. Revision E: Read-write. Specifies whether <a href="#">[The Max Performance Frequency Clock Count (MPERF)] MSR0000_00E7</a> and <a href="#">[The Actual Performance Frequency Clock Count (APERF)] MSR0000_00E8</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="#">2.4.5 [Effective Frequency]</a> for details.
25	<b>CpbDis: core performance boost disable.</b> Revision D and earlier: Reserved. Revision E: Read-write. 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="#">2.4.2.1.1 [Core Performance Boost (CPB)]</a> . If core performance boost is disabled while the core is in a boosted P-state, the core will automatically transition to the highest performance non-boosted P-state.
24	<b>TscFreqSel: TSC frequency select.</b> Read-write. 0=The TSC increments at the rate of the NCLK frequency; the TSC increments by 1 while in NB P-state 0 and increments by 2 while in NB P-state 1. 1=The TSC increments at the rate specified by the core P0 frequency at the time this bit is set by software. This field uses software P-state numbering. See <a href="#">2.4.2.1.2.1 [Software P-state Numbering]</a> . Changing the state of this bit after setting it results in undefined behaviour from the TSC. Changing the P0 frequency after setting this bit has no effect on the TSC rate. BIOS should program this bit to 1.
23	<b>ForceUsRdWrSzPrb: force probes for upstream RdSized and WrSized.</b> Read-write. 1=Forces probes on all upstream read-sized and write-sized transactions except for display refresh transactions. This bit is shared between all cores in a node.
22	Reserved.
21	<b>MisAlignSseDis: misaligned SSE mode disable.</b> Read-write. 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. 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="#">[The IO Trap Control Register (SMI_ON_IO_TRAP_CTL_STS)] MSRC001_0054</a> .
19	Reserved.

18	<p><b>McStatusWrEn: machine check status write enable.</b> Read-write. 1=Writes by software to MCI_STATUS (see 2.12.1 [Machine Check Architecture]) do not cause general protection faults; such writes update all implemented bits in these registers. 0=Writing a non-zero pattern to these registers causes a general protection fault. This also affects bits in [The NB Machine Check Misc (Thresholding) Registers] F3x1[78, 70, 68, 60].</p> <p>McStatusWrEn can be used to debug machine check interrupt handlers. See 2.12.1.8 [Error Injection and Simulation].</p>
17	<p><b>Wrap32Dis: 32-bit address wrap disable.</b> Read-write. 1=Disable 32-bit address wrapping. Software can use Wrap32Dis to access physical memory above 4 Gbytes without switching into 64-bit mode. To do so, software should write a greater-than 4 Gbyte address to [The FS Base Register (FS_BASE)] MSRC000_0100 and [The GS Base Register (GS_BASE)] MSRC000_0101. Then it would address <math>\pm 2</math> 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.</p>
16	Reserved.
15	<p><b>SseDis: SSE instructions disable.</b> Read-write. 1=Disables SSE instructions. If this is set, then CPUID Fn[8000_0001, 0000_0001]_EDX[SSE, SSE2], CPUID Fn0000_0001_ECX[SSE3], and CPUID Fn8000_0001_ECX[SSE4A] are 0.</p>
14	<p><b>RsmSpCycDis: RSM special bus cycle disable.</b> Read-write; read-only if SmmLock=1. 0=A link special bus cycle, SMIACK, is generated on a resume from SMI.</p>
13	<p><b>SmiSpCycDis: SMI special bus cycle disable.</b> Read-write; read-only if SmmLock=1. 0=A link special bus cycle, SMIACK, is generated when an SMI interrupt is taken.</p>
12	<p><b>HltXSpCycEn: halt-exit special bus cycle enable.</b> Read-write. 1=A link special bus cycle is generated when exiting from the halt state.</p>
11	<p><b>LimitCpuidStdMaxVal.</b> Read-write. 1=Limit CPUID standard maximum value, returned by CPUID Fn0000_0000_EAX (see CPUID Fn[8000_0000, 0000_0000]), to 1, and forces all standard functions greater than 1 to 0.</p>
10	<p><b>MonMwaitUserEn: MONITOR/MWAIT user mode enable.</b> Read-write. 1=The MONITOR and MWAIT instructions are supported in all privilege levels. 0=The MONITOR and MWAIT instructions are supported only in privilege level 0; these instructions in privilege levels 1 to 3 cause a #UD exception. The state of this bit is ignored if MonMwaitDis is set.</p>
9	<p><b>MonMwaitDis: MONITOR and MWAIT disable.</b> Read-write. 1=The MONITOR and MWAIT opcodes become invalid. This affects what is reported back through CPUID Fn0000_0001_ECX[Monitor].</p>
8	<p><b>IgnneEm: IGNNE port emulation enable.</b> Read-write. 1=Enable emulation of IGNNE port.</p>
7:5	Reserved.
4	<p><b>INVD_WBINVD: INVD to WBINVD conversion.</b> Read-write. 1=Convert INVD to WBINVD. This bit is required to be set when the L3 cache is enabled.</p>
3	<p><b>TlbCacheDis: cacheable memory disable.</b> Read-write. 0=Enables performance optimization that assumes PML4, PDP, PDE, and PTE entries are in cacheable WB-DRAM; memory type checks may be bypassed, and addresses outside of WB-DRAM may result in undefined behavior or NB protocol errors. 1=Disables performance optimization and allows PML4, PDP, PDE and PTE entries to be in any memory type. Operating systems that maintain page tables in memory types other than WB-DRAM must set TlbCacheDis to insure proper operation.</p>
2	Reserved.

1	<b>SlowFence: slow SFENCE enable.</b> Read-write. 1=Enable slow sfence.
0	<b>SmmLock: SMM code lock.</b> Read; write-1-only. 1=SMM configuration registers SMM_BASE, SMMAddr, SMMMask (all except for SMMMask[TClose:AClose]), and SMM_CTL are read-only and SMI interrupts are not intercepted in SVM.

### MSRC001\_00[18, 16] IO Range Registers Base (IORR\_BASE[1:0])

Reset: X. MSRC001\_0016 and MSRC001\_0017 combine to specify the first IORR range and MSRC001\_0018 and MSRC001\_0019 combine to specify the second IORR range. A CPU access--with address CPUAddr--is determined to be within IORR address range if the following equation is true:  
 $CPUAddr[47:12] \& PhyMask[47:12] == PhyBase[47:12] \& PhyMask[47:12]$ .

Bits	Description
63:48	RAZ.
47:12	<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 Registers Mask (IORR\_MASK[1:0])

Reset: X. See [MSRC001\\_00\[18, 16\]](#).

Bits	Description
63:48	RAZ.
47:12	<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 Register (TOP\_MEM)

Reset: X.

Bits	Description
63:48	RAZ.
47:23	<b>TOM[47:23]: 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="#">2.9.3 [Access Type Determination]</a> .
22:0	RAZ.

### MSRC001\_001D Top Of Memory 2 Register (TOM2)

Reset: X.

Bits	Description
63:48	RAZ.
47:23	<b>TOM2[47:23]: 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 2.9.3 [Access Type Determination]. This register is enabled by [The System Configuration Register (SYS_CFG)] MSRC001_0010[MtrrTom2En].
22:0	RAZ.

### MSRC001\_001F Northbridge Configuration Register (NB\_CFG)

Reset: 0000 0000 0000 0008h. Software is required to perform a read-modify-write in order to change any of the values in this register. This register is accessible through F3x[8C:88] as well. Only one of these registers exists in multi-core devices; see 3.1.1 [Northbridge MSRs In Multi-Core Products].

Bits	Description
63	Reserved
62	<b>GfxNbPstateDis.</b> Revisions C1 and earlier, RB-C2, BL-C2, and D and later: Reserved. Revision DA-C and C3: Read-write. 1=Disables NB P-state transitions and changes specified by MSRC001_00[68:64][NbVid, NbDid] and MSRC001_0061[CurPstateLimit]. 0=Enables NB P-state transitions and changes specified by MSRC001_00[68:64][NbVid, NbDid] and MSRC001_0061[CurPstateLimit]. See 2.4.2.2.1 [NB P-state Control] and 2.7.10 [Miscellaneous Behaviors and Requirements]. For software initiated self-refresh, this bit must be set prior to setting EnterSelfRef and restored after the subsequent ExitSelfRef completes.
61:55	Reserved.
54	<b>InitApicIdCpuIdLo.</b> Read-write. Revision C and earlier: 0=Initial value of APIC20[ApicId[7:0]] is {CpuCoreNum[1:0], 000b, F0x60[NodeId[2:0]]}. 1=Initial value of APIC20[ApicId[7:0]] is {000b, F0x60[NodeId[2:0]], CpuCoreNum[1:0]}. Revision D and later single-node processors: 0=Initial value of APIC20[ApicId[7:0]] is {CpuCoreNum[2:0], 00b, F0x60[NodeId[2:0]]}. 1=Initial value of APIC20[ApicId[7:0]] is {00b, F0x60[NodeId[2:0]], CpuCoreNum[2:0]}. Revision D and later dual-node processors: 0=Initial value of APIC20[ApicId[7:0]] is {CpuCoreNum[3:0], 00b, F0x60[NodeId[2:1]]}. 1=Initial value of APIC20[ApicId[7:0]] is {00b, F0x60[NodeId[2:1]], CpuCoreNum[3:0]}. See 2.9.2 [CPU Cores and Downcoring] for information about CpuCoreNum. This bit should always be set by BIOS; it should be set before F0x60[NodeId] is programmed.
53:51	Reserved.
50	<b>DisOrderRdRsp.</b> Read-write. 1=Disables ordered responses to IO link read requests. See 2.7.8 [Response Ordering].
49:47	Reserved.
46	<b>EnableCf8ExtCfg: enable CF8 extended configuration cycles.</b> Read-write. 1=Allows the IO configuration space access method, IOCF8 and IOCFC, to be used to generate extended configuration cycles by enabling IOCF8[27:24].
45	<b>DisUsSysMgtReqToNcHt: disable upstream system management request to link.</b> Read-write. 1=Disables downstream reflection of upstream STPCLK and x86 legacy input system management commands (in order to work around potential deadlock scenarios related to reflection regions).

44:37	Reserved
36	<b>DisDatMsk: disable data mask.</b> Read-write. 1=Disables DRAM data masking function; all write requests that are less than one cacheline, a DRAM read is performed before writing the data.
35:32	Reserved
31	<b>DisCohLdtCfg: disable coherent link configuration accesses.</b> Read-write. 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 [The Configuration Map Registers] F1x[EC:E0]. 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.
30:11	Reserved.
10	<b>DisXdsBypass: disable xbar data scheduler bypass.</b> Read-write. 1=The crossbar data scheduler bypass is disabled. This bit should be set in systems containing coherent devices that are not AMD Family 10h processors.
9	<b>DisRefUseFreeBuf: disable display refresh to use free list buffers.</b> Read-write. 1=In non-IFCM disable display refresh requests from using free list buffers and in IFCM disable isochronous requests from using free list buffers.
8:0	Reserved.

### MSRC001\_0022 Machine Check Exception Redirection Register

Reset: 0000 0000 0000 0000h. This register can be used to redirect machine check exceptions (MCEs) to SMIs or vectored interrupts. If both RedirSmiEn and RedirVecEn are set, then undefined behavior results.

Bits	Description
63:10	Reserved.
9	<b>RedirSmiEn.</b> Read-write. 1=Redirect MCEs (that are directed to this core) to generate an SMI-trigger IO cycle via MSRC001_0056. The status is stored in SMMFEC4[MceRedirSts].
8	<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 Registers

Reset: 0000 0000 0000 0000h. These registers holds the CPUID name string in ASCII. The state of these registers are returned by CPUID instructions, CPUID Fn8000\_000[4, 3, 2]. BIOS should set these registers to the product name for the processor as provided by AMD. Each register contains a block of 8 ASCII characters; the least byte corresponds to the first ASCII character of the block; the most-significant byte corresponds to the last character of the block. MSRC001\_0030 contains the first block of the name string; MSRC001\_0035 contains the last block of the name string.

Bits	Description
63:0	<b>CpuNameString.</b> Read-write.

### **MSRC001\_003E Hardware Thermal Control Register (HTC)**

See [The Hardware Thermal Control (HTC) Register] F3x64. Only one of these registers exists in multi-core devices; see 3.1.1 [Northbridge MSRs In Multi-Core Products].

### **MSRC001\_00[49:44] Machine Check Control Mask Registers (MCi\_CTL\_MASK)**

Reset:	MSRC001_0044: 0000 0000 0000 0080h.	MSRC001_0047: 0000 0000 0000 0000h.
	MSRC001_0045: 0000 0000 0000 0080h.	MSRC001_0048: 0000 0000 0000 0000h.
	MSRC001_0046: 0000 0000 0000 0200h.	MSRC001_0049: 0000 0000 0000 0000h.

Regarding MSRC001\_0048, only one of these registers exists in multi-core devices; see 3.1.1 [Northbridge MSRs In Multi-Core Products].

BIOS is recommended to mask HT retries by setting the bits in MSRC001\_0048 corresponding to F3x40[RtryHt3En, RtryHt2En, RtryHt1En, RtryHt0En] if the OS is not capable of distinguishing that HT retries are normal operation.

BIOS is recommended to mask GART table walk errors by setting the bit in MSRC001\_0048 corresponding to F3x40[GartTblWkEn].

These mask registers should be set up prior to enabling errors in MCi\_CTL registers. BIOS must not clear MSK bits that are reset to 1.

Bits	Description
63:0	<b>MSK: Control Register Masks.</b> Bits are read-only or read-write, corresponding to the attribute of the same bit in MCi_CTL. 1=Disable error logging in MCi_STATUS and MCi_ADDR for errors represented by the corresponding bit in MCi_CTL. See 2.12.1 [Machine Check Architecture]. Disabling logging is equivalent to disabling error detection, and prevents error responses.

### **MSRC001\_00[53:50] IO Trap Registers (SMI\_ON\_IO\_TRAP\_[3:0])**

Reset: 0000 0000 0000 0000h. MSRC001\_00[53:50] and MSRC001\_0054 provide a mechanism for executing the SMI handler if a an access to one of the specified addresses is detected. Access address and access type checking is done before IO instruction execution. If the access address and access type match one of the specified IO address and access types, then: (1) the IO instruction is not executed; (2) any breakpoint, other than the single-step breakpoint, set on the IO instruction is not taken (the single-step breakpoint is taken after resuming from SMM); and (3) the SMI-trigger IO cycle specified by MSRC001\_0056. The status is stored in SMMFEC4[IoTrapSts].

IO-space configuration accesses are special IO accesses. An IO access is defined as an IO-space configuration access when IO instruction address bits[31:0] are CFCh, CFDh, CFEh, or CFFh when IO-space configuration is enabled (IOCF8[ConfigEn]). The access address for a configuration space access is the current value of IOCF8[BusNo, Device, Function, RegNo]. The access address for an IO access that is not a configuration access is equivalent to the IO instruction address, bits[31:0].

The access address is compared with SmiAddr, and the instruction access type is compared with the enabled access types defined by ConfigSMI, SmiOnRdEn, and SmiOnWrEn. Access address bits[23:0] can be masked with SmiMask.

IO and configuration space trapping to SMI applies only to single IO instructions; it does not apply to string



and REP IO instructions.

Bits	Description
63	<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 Register (SMI\_ON\_IO\_TRAP\_CTL\_STS)

Reset: 0000 0000 0000 0000h. For each of the SmiEn bits below, 1=The trap specified by the corresponding MSR is enabled. See [MSRC001\\_00\[53:50\]](#).

Bits	Description
63:32	RAZ.
31:16	Reserved.
15	<b>IoTrapEn: IO trap enable.</b> Read-write. 1=Enable IO and configuration space trapping specified by <a href="#">MSRC001_00[53:50]</a> and <a href="#">MSRC001_0054</a> .
14:8	Reserved.
7	<b>SmiEn_3: SMI enable for the trap specified by MSRC001_0053.</b> Read-write.
6	Reserved.
5	<b>SmiEn_2: SMI enable for the trap specified by MSRC001_0052.</b> Read-write.
4	Reserved.
3	<b>SmiEn_1: SMI enable for the trap specified by MSRC001_0051.</b> Read-write.
2	Reserved.
1	<b>SmiEn_0: SMI enable for the trap specified by MSRC001_0050.</b> Read-write.
0	Reserved.

### MSRC001\_0055 Interrupt Pending and CMP-Halt Register

Reset: 0000 0000 0000 0000h. This register is used to specify messages that the processor generates under certain conditions, that target the IO hub. One purpose is to ensure that the IO hub can wake the processor out of the stop-grant state when there is a pending interrupt. Otherwise, it is possible for the processor to remain in the stop-grant state while an interrupt is pending in the processor. This is accomplished by sending a message to the IO hub to indicate that the interrupt is pending. There are two message types: a programmable IO-space message and the link INT\_PENDING message defined by the link specification.

If the IO hub does not support the INT\_PENDING message, the IO space message should be selected by IntPndMsg. When this is enabled, the check for a pending interrupt is performed at the end of each IO instruction. If there is a pending interrupt and STPCLK is asserted, the processor executes a byte-size IO access as specified by IORd, IOMsgAddr, and IOMsgData.

If the IO hub supports the INT\_PENDING message, it should be selected by IntPndMsg. The check for a pend-



ing interrupt is performed while in the stop-grant state or when entering the stop-grant state. If there is a pending interrupt, the processor broadcasts the INT\_PENDING message. An INT\_PENDING message may not be generated for arbitrated interrupts in multi-node systems.

Bits	Description
63:32	RAZ.
31:30	Reserved.
29	<b>BmStsClrOnHltEn: BM_STS clear on halt enable.</b> Revision D0 and earlier: Reserved. Revision D1 and later: Read-write. 1=The core clears BM_STS in the IO Hub after caches are flushed upon execution of the HLT instruction by writing 0010h to the port address specified by IOMsgAddr. This bit has no effect if F3xDC[CacheFlushOnHaltCtl]=0. This bit should be clear if F3xD4[MTC1eEn]=0. See 2.4.3.3.5 [BIOS Requirements to Initialize Message Triggered C1E].
28	<b>C1eOnCmpHalt: C1E on chip multi-processing halt.</b> Read-write. 1=When all cores of a processor have entered the halt state, the processor generates an IO cycle as specified by IORd, IOMsgData, and IOMsgAddr. When this bit is set, SmiOnCmpHalt and IntrPndMsg must be 0, otherwise the behavior is undefined. For revision DA-C, this bit is only supported for dual-core processors. For revision C3 and E, this bit is supported for any number of cores. See 2.4.3.3.3 [Hardware Initiated C1E].
27	<b>SmiOnCmpHalt: SMI on chip multi-processing halt.</b> Read-write. 1=When all cores of the processor have entered the halt state, the processor generates an SMI-trigger IO cycle as specified by IORd, IOMsgData, and IOMsgAddr. When this bit is set C1eOnCmpHalt and IntrPndMsg must be 0, otherwise the behavior is undefined. The status is stored in SMMFEC4[SmiOnCmpHaltSts]. See 2.4.3.3.1 [SMI Initiated C1E].
26	<b>IORd: IO Read.</b> Read-write. 1=IO read; 0=IO write.
25	<b>IntrPndMsg: interrupt pending message.</b> Read-write. Revision C2 and earlier: Selects the interrupt pending message type. 0=Link-defined INT_PENDING message; 1=Programmable SMI-trigger IO-space message. The status is stored in SMMFEC4[IntPendSmiSts]. Revision C3 and later: Reserved.
24	<b>IntrPndMsgDis: interrupt pending message disable.</b> Read-write. Revision C2 and earlier: Disable generating the interrupt pending message specified by IntrPndMsg. Revision C3 and later: Reserved.
23:16	<b>IOMsgData: IO message data.</b> Read-write. IO write message data. This field is only used if IORd specifies an IO write message.
15:0	<b>IOMsgAddr: IO message address.</b> Read-write. IO space message address.

### MSRC001\_0056 SMI Trigger IO Cycle Register

Reset: 0000 0000 0000 0000h. See 2.13.2.3 [SMI Sources And Delivery]. This register specifies an IO cycle that may be generated when a local SMI trigger event occurs. If IoCycleEn is set and there is a local SMI trigger event, then the IO cycle generated is a byte read or write, based on IoRd, to address IoPortAddress. If the cycle is a write, then IoData contains the data written. If the cycle is a read, the value read is discarded. If IoCycleEn is clear and a local SMI trigger event occurs, then undefined behavior results.

Bits	Description
63:27	Reserved.
26	<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 Register

Reset: xxxx xxxx xxxx xxx0h. See 2.11 [Configuration Space] for a description of MMIO configuration space. All cores of all processors should be programmed with the same value of this register.

Bits	Description																																				
63:48	RAZ.																																				
47:20	<b>MmioCfgBaseAddr[47:20]: MMIO configuration base address bits[47:20].</b> Read-write. Specifies the base address of the MMIO configuration range. The size of the MMIO configuration-space address range is specified by BusRange.																																				
19:6	RAZ.																																				
5:2	<b>BusRange:bus range identifier.</b> Read-write. This specifies the number of busses in the MMIO configuration space range. The size of the MMIO configuration space is 1 Mbyte times the number of busses. This field is encoded as follows: <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Bits</th> <th>Buses</th> <th>Bits</th> <th>Buses</th> </tr> </thead> <tbody> <tr> <td>0h</td> <td>1</td> <td>8h</td> <td>256</td> </tr> <tr> <td>1h</td> <td>2</td> <td>9h</td> <td>Reserved</td> </tr> <tr> <td>2h</td> <td>4</td> <td>Ah</td> <td>Reserved</td> </tr> <tr> <td>3h</td> <td>8</td> <td>Bh</td> <td>Reserved</td> </tr> <tr> <td>4h</td> <td>16</td> <td>Ch</td> <td>Reserved</td> </tr> <tr> <td>5h</td> <td>32</td> <td>Dh</td> <td>Reserved</td> </tr> <tr> <td>6h</td> <td>64</td> <td>Eh</td> <td>Reserved</td> </tr> <tr> <td>7h</td> <td>128</td> <td>Fh</td> <td>Reserved</td> </tr> </tbody> </table>	Bits	Buses	Bits	Buses	0h	1	8h	256	1h	2	9h	Reserved	2h	4	Ah	Reserved	3h	8	Bh	Reserved	4h	16	Ch	Reserved	5h	32	Dh	Reserved	6h	64	Eh	Reserved	7h	128	Fh	Reserved
Bits	Buses	Bits	Buses																																		
0h	1	8h	256																																		
1h	2	9h	Reserved																																		
2h	4	Ah	Reserved																																		
3h	8	Bh	Reserved																																		
4h	16	Ch	Reserved																																		
5h	32	Dh	Reserved																																		
6h	64	Eh	Reserved																																		
7h	128	Fh	Reserved																																		
1	Reserved.																																				
0	<b>Enable.</b> 1=MMIO configuration space is enabled.																																				

### MSRC001\_0061 P-State Current Limit Register

See 2.4.2 [P-states]. Writes to this register cause a #GP.

Bits	Description
63:7	RAZ.
6:4	<b>PstateMaxVal: P-state maximum value.</b> Read-only. Specifies the lowest performance non-boosted P-state (highest non-boosted value) supported by the hardware. This field uses software P-state numbering. See 2.4.2.1.2.1 [Software P-state Numbering]. The state of this field is controlled through [The Clock Power/Timing Control 2 Register] F3xDC[PstateMaxVal].
3	RAZ.
2:0	<b>CurPstateLimit: current P-state limit.</b> Read-only. Specifies the highest-performance non-boosted P-state (lowest value) allowed. CurPstateLimit is always bounded by MSRC001_0061[PstateMaxVal]. Attempts to change CurPstateLimit to a value greater (lower performance) than MSRC001_0061[PstateMaxVal] leaves CurPstateLimit unchanged. See MSRC001_0071[CurPstateLimit] and MSRC001_001F[GfxNbPstateDis]. This field uses software P-state numbering. See 2.4.2.1.2.1 [Software P-state Numbering].

**MSRC001\_0062 P-State Control Register**

Bits	Description
63:3	MBZ.
2:0	<b>PstateCmd: P-state change command.</b> Read-write. Cold reset: values vary 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[68:64]</a> . 0=P-state 0; 1=P-state 1; ... 4=P-state 4. Values of 5h through 7h are reserved. P-state limits are applied appropriately. See <a href="#">2.4.2 [P-states]</a> . 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="#">2.4.2 [P-states]</a> and <a href="#">2.4.2.1.2.1 [Software P-state Numbering]</a> .

**MSRC001\_0063 P-State Status Register**

Writes to this register cause a #GP.

Bits	Description
63:3	RAZ.
2:0	<b>CurPstate: current P-state.</b> Read-only. Cold reset: values vary 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> and <a href="#">[The SBI P-state Limit Register] F3xC4</a> ; see <a href="#">2.4.2.5 [P-state Transition Behavior]</a> for information on how these interact). 0=P-state 0; 1=P-state 1; 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="#">2.4.2 [P-states]</a> and <a href="#">2.4.2.1.2.1 [Software P-state Numbering]</a> .

**MSRC001\_00[68:64] P-State [4:0] Registers**

Reset: values vary by product as specified by [F4x1\[F0:E0\]](#).

Each of these registers specify the frequency and voltage associated with each of the core.

- MSRC001\_0064 specifies P-state 0
- MSRC001\_0065 specifies P-state 1
- MSRC001\_0066 specifies P-state 2
- MSRC001\_0067 specifies P-state 3
- MSRC001\_0068 specifies P-state 4

The CpuVid and NbVid fields in these registers are 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. The only exception is CpuFid of the boosted P-states for boost enabled processors may differ among cores. See [2.4.2 \[P-states\]](#) for more information about these registers.

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	Reserved.
41:40	<b>IddDiv: current divisor field.</b> Read-write. See <a href="#">MSRC001_00[68:64][IddValue]</a> .

39:32	<p><b>IddValue: current value field.</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. For boost P-states, IddDiv and IddValue combine to specify the effective maximum current dissipation of a single core in the boost P-state. These values are intended to be used to create ACPI-defined _PSS objects (see 2.4.2.13 [ACPI Processor P-State Objects]) and to perform the 2.4.2.11 [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, may be subsequently altered by software; they do not affect the hardware behavior. These fields are encoded as follows:</p> <table border="1"> <thead> <tr> <th><u>IddDiv</u></th> <th><u>Current Equation</u></th> <th><u>Current Range</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>IddValue / 1 A</td> <td>0 to 255 A</td> </tr> <tr> <td>01b</td> <td>IddValue / 10 A</td> <td>0 to 25.5 A</td> </tr> <tr> <td>10b</td> <td>IddValue / 100 A</td> <td>0 to 2.55 A</td> </tr> <tr> <td>11b</td> <td>Reserved</td> <td></td> </tr> </tbody> </table>	<u>IddDiv</u>	<u>Current Equation</u>	<u>Current Range</u>	00b	IddValue / 1 A	0 to 255 A	01b	IddValue / 10 A	0 to 25.5 A	10b	IddValue / 100 A	0 to 2.55 A	11b	Reserved	
<u>IddDiv</u>	<u>Current Equation</u>	<u>Current Range</u>														
00b	IddValue / 1 A	0 to 255 A														
01b	IddValue / 10 A	0 to 25.5 A														
10b	IddValue / 100 A	0 to 2.55 A														
11b	Reserved															
31:25	<p><b>NbVid: Northbridge VID.</b> Read-write. See 2.4.1 [Processor Power Planes And Voltage Control]. This field is required to be programmed as specified by MSRC001_0071 [MaxVid and MinVid] (otherwise undefined behavior results). In SVI platforms, the value of this field must be the same in all MSRs in which NbDid=0; the value of this field must be the same in all MSRs in which NbDid=1. In PVI platforms, this may vary with each CPU P-state, as NbVid specifies the voltage of the NB and cores. See F3x1F0[NbPstate].</p>															
24:23	Reserved.															
22	<p><b>NbDid: Northbridge divisor ID.</b> Read-write. Specifies the NB frequency divisor; see F3xD4[NbFid]. 0=Divisor of 1 (NB P-state 0). 1=Divisor of 2 (NB P-state 1). 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 2.4.2.2 [NB P-states] and F3x1F0[NbPstate].</p>															
21:16	Reserved.															
15:9	<p><b>CpuVid: core VID.</b> Read-write. See 2.4.1 [Processor Power Planes And Voltage Control]. This field is required to be programmed as specified by MSRC001_0071 [MaxVid and MinVid] (otherwise undefined behavior results).</p>															
8:6	<p><b>CpuDid: core divisor ID.</b> Read-write. Specifies the CPU frequency divisor; see CpuFid.</p> <table border="1"> <tbody> <tr> <td>0h=Divisor of 1</td> <td>3h=Divisor of 8</td> </tr> <tr> <td>1h=Divisor of 2</td> <td>4h=Divisor of 16</td> </tr> <tr> <td>2h=Divisor of 4</td> <td>5h - 7h=Reserved</td> </tr> </tbody> </table>	0h=Divisor of 1	3h=Divisor of 8	1h=Divisor of 2	4h=Divisor of 16	2h=Divisor of 4	5h - 7h=Reserved									
0h=Divisor of 1	3h=Divisor of 8															
1h=Divisor of 2	4h=Divisor of 16															
2h=Divisor of 4	5h - 7h=Reserved															
5:0	<p><b>CpuFid: core frequency ID.</b> Read-write. Specifies the CPU frequency multiplier. The CPU COF specified by CpuFid and CpuDid is: <math>\text{CPU COF} = 100 \text{ MHz} * (\text{CpuFid} + 10h) / (2^{\text{CpuDid}})</math>. This field and CpuDid must be programmed to the requirements specified in MSRC001_0071 [MaxCpuCof] and F3x1F0 [MaxSwPstateCpuCof]. If F4x15C [NumBoostStates]=1, writes to a non-boosted P-state with frequencies greater than F3x1F0 [MaxSwPstateCpuCof] are ignored. The value of this field must be less than or equal to 2Fh.</p>															

### MSRC001\_0070 COFVID Control Register

Cold reset: values vary by product.

This register includes several fields that are identical to MSRC001\_00 [68:64]. 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 2.4.2 [P-states].

Bits	Description
63:32	RAZ.
31:25	<b>NbVid: Northbridge VID.</b> See <a href="#">MSRC001_00[68:64][NbVid]</a> .
24:23	RAZ.
22	<b>NbDid: Northbridge divisor ID.</b> See <a href="#">MSRC001_00[68:64][NbDid]</a> .
21:19	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="#">2.4.2.1.2.2 [Hardware P-state Numbering]</a> .
15:9	<b>CpuVid: core VID.</b> Read-write. See <a href="#">MSRC001_00[68:64]</a> .
8:6	<b>CpuDid: core divisor ID.</b> Read-write. See <a href="#">MSRC001_00[68:64]</a> . The PstateId field must be updated to cause a new CpuDid value to take effect.
5:0	<b>CpuFid: core frequency ID.</b> Read-write. See <a href="#">MSRC001_00[68:64]</a> . The PstateId field must be updated to cause a new CpuFid value to take effect.

### MSRC001\_0071 COFVID Status Register

See [2.4.2 \[P-states\]](#).

Bits	Description
63:59	<b>MaxNbFid: maximum NB COF.</b> Read-only. Specifies the maximum NB FID supported by the processor. The maximum frequency is 200 MHz * (MaxNbFid + 4), if MaxNbFid is greater than zero; if MaxNbFid = 00h, then there is no frequency limit. Any attempt to change the NB FID to a frequency greater than specified by this register results in no change to the NB FID.
58:56	<b>CurPstateLimit: current P-state limit.</b> Read-only. Specifies the current highest-performance P-state limit (lowest value) allowed. This register uses hardware P-state numbering. See <a href="#">MSRC001_0061[CurPstateLimit]</a> and <a href="#">2.4.2.1.2.2 [Hardware P-state Numbering]</a> .
55	Reserved.
54:49	<b>MaxCpuCof: maximum CPU COF.</b> Read-only. 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.
48:42	<b>MinVid: minimum voltage.</b> Read-only. Specifies the VID code corresponding to the minimum voltage that the processor drives. 00h indicates that no minimum VID code is specified. See <a href="#">2.4.1 [Processor Power Planes And Voltage Control]</a> .
41:35	<b>MaxVid: maximum voltage.</b> Read-only. Specifies the VID code corresponding to the maximum voltage that the processor drives. 00h indicates that no maximum VID code is specified. See <a href="#">2.4.1 [Processor Power Planes And Voltage Control]</a> .
34:32	<b>StartupPstate: startup P-state number.</b> Read-only. Specifies the cold reset VID, FID and DID for the NB and core based on the P-state number selected (see <a href="#">MSRC001_00[68:64]</a> ). StartupPstate uses hardware P-state numbering. If <a href="#">F3xA0[PviMode]=1</a> , then NbVid of the selected P-state is applied to the PVI. If <a href="#">F3xA0[CofVidProg]=0</a> , then the state of this field is ignored and the VID, FID, and DID are applied to the core and NB as specified by that bit.

31:25	<b>CurNbVid: current Northbridge VID.</b> Read-only.
24:23	Reserved.
22	<b>CurNbDid: current Northbridge divisor ID.</b> Read-only.
21:19	Reserved.
18:16	<b>CurPstate: current P-state.</b> Read-only. Specifies the current P-state requested by the core. This field uses hardware P-state numbering. See <a href="#">MSRC001_0063[CurPstate]</a> and <a href="#">2.4.2.1.2.2 [Hardware P-state Numbering]</a> .
15:9	<b>CurCpuVid: current core VID.</b> Read-only. In dual-node processors, CurCpuVid on internal node 0 is the voltage driven to VDD. CurCpuVid on internal node 1 is the voltage of the highest performance P-state (lowest numbered P-state) requested by all cores on internal node 1. CurCpuVid on internal node 1 is greater than or equal to CurCpuVid on internal node 0.
8:6	<b>CurCpuDid: current core divisor ID.</b> Read-only.
5:0	<b>CurCpuFid: current core frequency ID.</b> Read-only.

### MSRC001\_0072 SBI P-state Limit Register

This register is a duplicate of [\[The SBI P-state Limit Register\] F3xC4](#). Only one of these registers exists in multi-core devices; see [3.1.1 \[Northbridge MSRs In Multi-Core Products\]](#). This register is only accessible via SB-RMI. See [2.12.3 \[Sideband Interface \(SBI\)\]](#).

### MSRC001\_0073 C-state Base Address

Revision D and earlier: GP-read-write. Revision E: Reset: 0000\_0000\_0000\_0000h.

Bits	Description
63:16	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 results in undefined behavior. All other values cause the core to trap IO addresses CstateAddr through CstateAddr+7. BIOS should program this field to an available IO address. See <a href="#">2.4.3.2 [C-state Request Interface]</a> .

### MSRC001\_0074 CPU Watchdog Timer Register (CpuWdTmrCfg)

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\[OVER\]](#)), a sync flood can be generated if enabled in [F3x180\[SyncFloodOnCpuLeakErr\]](#).

The CPU watchdog timer must be set higher than the NB watchdog timer ([\[The MCA NB Configuration Register\] F3x44](#)) 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	<b>CpuWdtCountSel: CPU watchdog timer count select.</b> Read-write. This, along with CpuWdtTimeBase, specifies the time period required for the WDT to expire. The time period is the value specified here times the time base specified by CpuWdtTimeBase. Note that the actual timeout period may be anywhere from zero to one increments less than the values specified, due to non-deterministic behavior. The field is encoded as follows: 0000b = 4095                      0100b = 255                      1000b = 8191 0001b = 2047                      0101b = 127                      1001b = 16383 0010b = 1023                      0110b = 63                      1010b - 1111b Reserved. 0011b = 511                      0111b = 31
2:1	<b>CpuWdtTimeBase: CPU watchdog timer time base.</b> Read-write. Specifies the time base for the timeout period specified in CpuWdtCountSel. 00b = 1.31 ms.                      10b = 5 ns. 01b = 1.28 us.                      11b = Reserved.
0	<b>CpuWdtEn: CPU watchdog timer enable.</b> Read-write. 1=The WDT is enabled.

### MSRC001\_0111 SMM Base Address Register (SMM\_BASE)

Reset: 0000 0000 0003 0000h. This holds the base of the SMM memory region. The value of this register is stored in the save state on entry into SMM (see 2.13.2.5 [SMM Save State]) and it is restored on returning from SMM. The 16-bit CS (code segment) selector is loaded with SMM\_BASE[19:4] on entering SMM. SMM\_BASE[31:20] and SMM\_BASE[3:0] are 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 SMM\_BASE with the new value.
- Normal WRMSR access to this register.

Bits	Description
63:32	Reserved.
31:0	<b>SMM_BASE.</b> Read-write; read-only if MSRC001_0015[SmmLock]=1.

### MSRC001\_0112 SMM TSeg Base Address Register (SMMAddr)

Reset: 0000 0000 0000 0000h. See 2.13.2 [System Management Mode (SMM)] for information about SMM. 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:

$$\text{CPUAddr}[47:17] \& \text{TSegMask}[47:17] == \text{TSegBase}[47:17] \& \text{TSegMask}[47:17].$$

For example, if TSeg spans 256K bytes and starts at the 1M byte address. The MSRC001\_0112[TSegBase] would be set to 0010\_0000h and the MSRC001\_0113[TSegMask] to FFFC\_0000h (with zeros filling in for bits[16:0]). This results in a TSeg range from 0010\_0000 to 0013\_FFFFh.

Bits	Description
63:48	Reserved.
47:17	<b>TSegBase[47:17]: TSeg address range base.</b> Read-write; read-only if MSRC001_0015[SmmLock]=1.
16:0	Reserved.



**MSRC001\_0113 SMM TSeg Mask Register (SMMMMask)**

Reset: 0000 0000 0000 0000h. See 2.13.2 [System Management Mode (SMM)] for information about SMM.

The ASeg address range is located at a fixed address from A0000h–BFFFFh. The TSeg range is located at a variable base (specified by [MSRC001\\_0112](#)[TSegBase]) with a variable size (specified by [MSRC001\\_0113](#)[TSegMask]). These ranges provide a safe location for SMM code and data that is not readily accessible by non-SMM applications. The SMI handler can be located in one of these two ranges, or it can be located outside these ranges. These ranges must never overlap each other.

This register specifies how accesses to the ASeg and TSeg address ranges are control as follows:

- If [A, T]Valid=0, then the address range is accessed as specified by MTRRs, regardless of whether the CPU is in SMM or not.
- If [A, T]Valid=1, then:
  - If in SMM, then:
    - If [A, T]Close=0, then the accesses are directed to DRAM with memory type as specified in [A, T]MTypeDram.
    - If [A, T]Close=1, then instruction accesses are directed to DRAM with memory type as specified in [A, T]MTypeDram and data accesses are directed at MMIO space and with attributes based on [A, T]MTypeIoWc.
  - If not in SMM, then the accesses are directed at MMIO space with attributes based on [A, T]MTypeIoWc.

Bits	Description
63:48	Reserved.
47:17	<b>TSegMask[47:17]: TSeg address range mask.</b> Read-write; read-only if <a href="#">MSRC001_0015</a> [SmmLock]=1. See <a href="#">MSRC001_0112</a> .
16:15	Reserved.
14:12	<b>TMTypeDram: TSeg address range memory type.</b> Read-write; read-only if <a href="#">MSRC001_0015</a> [SmmLock]=1. Specifies the memory type for SMM accesses to the TSeg range that are directed to DRAM. The encoding is identical to the three LSBs of the MTRRs. See <a href="#">MSR0000_02</a> [0F:00].
11	Reserved.
10:8	<b>AMTypeDram: ASeg Range Memory Type.</b> Read-write; read-only if <a href="#">MSRC001_0015</a> [SmmLock]=1. Specifies the memory type for SMM accesses to the ASeg range that are directed to DRAM. The encoding is identical to the three LSBs of the MTRRs. See <a href="#">MSR0000_02</a> [0F:00].
7:6	Reserved.
5	<b>TMTypeIoWc: non-SMM TSeg address range memory type.</b> Read-write; read-only if <a href="#">MSRC001_0015</a> [SmmLock]=1. 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> Read-write; read-only if <a href="#">MSRC001_0015</a> [SmmLock]=1. 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 <a href="#">AClose</a> .

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> Read-write; read-only if <a href="#">MSRC001_0015</a> [SmmLock]=1. 1=The TSeg address range SMM enabled.
0	<b>AValid: enable ASeg SMM address range.</b> Read-write; read-only if <a href="#">MSRC001_0015</a> [SmmLock]=1. 1=The ASeg address range SMM enabled.

### MSRC001\_0114 Virtual Machine Control Register (VM\_CR)

Reset: 0000 0000 0000 0000h.

Bits	Description
63:32	Reserved.
31:5	MBZ.
4	<b>Svme_Disable: SVME disable.</b> See Lock. 1= <a href="#">MSRC000_0080</a> [SVME] must be zero (MBZ) when writing to <a href="#">MSRC000_0080</a> . Setting this bit when <a href="#">MSRC000_0080</a> [SVME]=1 causes a #GP fault, regardless of the state of Lock. 0= <a href="#">MSRC000_0080</a> [SVME] is read-write.
3	<b>Lock: SVM lock.</b> Read-only; write-1-only (see <a href="#">MSRC001_0118</a> ). 1=Svme_Disable is read-only. 0=Svme_Disable is read-write.
2	<b>dis_a20m: disable A20 masking.</b> Read-write; set-by-hardware. 1=Disables A20 masking. This bit is set by hardware when the SKINIT instruction is executed.
1	<b>r_init: intercept INIT.</b> Read-write; set-by-hardware. This bit controls how INIT is delivered in host mode. This bit is set by hardware when the SKINIT instruction is executed. 0 = INIT delivered normally. 1 = INIT translated into a SX interrupt.
0	<b>dpd: debug port disable.</b> Read-write; set-by-hardware. This bit controls if debug facilities have access to the processor state information. This bit is set by hardware when the SKINIT instruction is executed. 0 = Debug port may be enabled. 1 = Debug port disabled; all mechanisms that could expose trusted code execution are disabled.

### MSRC001\_0115 IGNNE Register (IGNNE)

Bits	Description
63:1	MBZ.
0	<b>IGNNE: current IGNNE state.</b> Read-write. Reset: X. This bit controls the current state of the processor internal IGNNE signal.

### MSRC001\_0116 SMM Control Register (SMM\_CTL)

If [MSRC001\\_0015](#)[SmmLock]=1 then GP-read-write else GP-read. The bits in this register are processed in

the order of: `smm_enter`, `smi_cycle`, `smm_dismiss`, `rsm_cycle` and `smm_exit`. However, only the following combination of bits may be set in a single write (all other combinations result in undefined behavior):

- `smm_enter` and `smi_cycle`.
- `smm_enter` and `smm_dismiss`.
- `smm_enter`, `smi_cycle` and `smm_dismiss`.
- `smm_exit` and `rsm_cycle`.

Software is responsible for ensuring that `smm_enter` and `smm_exit` operations are properly matched and are not nested.

Bits	Description
63:5	MBZ.
4	<b>rsm_cycle: send RSM special cycle.</b> Write-only. Reset: X. 1=Send a RSM special cycle.
3	<b>smm_exit: exit SMM.</b> Write-only. Reset: X. 1=Exit SMM.
2	<b>smi_cycle: send SMI special cycle.</b> Write-only. Reset: X. 1=Send a SMI special cycle.
1	<b>smm_enter: enter SMM.</b> Write-only. Reset: X. 1=Enter SMM.
0	<b>smm_dismiss: clear SMI.</b> Write-only. Reset: X. 1=Clear the SMI pending flag.

#### MSRC001\_0117 Virtual Machine Host Save Physical Address Register (VM\_HSAVE\_PA)

Reset: 0000 0000 0000 0000h.

Bits	Description
63:0	<b>VM_HSAVE_PA: physical address of host save area.</b> Read-write. 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 $(FF\_FFFF\_Fh \geq VM\_HSAVE\_PA \geq FD\_0000\_0h)$ .

#### MSRC001\_0118 SVM Lock Key

Reset: 0000 0000 0000 0000h.

Bits	Description
63:0	<b>SvmLockKey: SVM lock key.</b> RAZ, write-only. Writes to this register when <code>MSRC001_0114[Lock]=0</code> write the <code>SvmLockKey</code> . Writes to this register when <code>MSRC001_0114[Lock]=1</code> and <code>SvmLockKey!=0</code> cause hardware to clear <code>MSRC001_0114[Lock]</code> if the value written is the same as the value stored in <code>SvmLockKey</code> .

#### MSRC001\_011A Local SMI Status

Reset: 0000 0000 0000 0000h. This registers returns the same information that is returned in [\[The Local SMI Status\] SMMFEC4](#) portion of the SMM save state. The information in this register is only updated when `MSRC001_0116[smm_dismiss]` is set by software.

Bits	Description
63:32	Reserved.
31:0	See <a href="#">[The Local SMI Status] SMMFEC4</a> .

**MSRC001\_0140 OS Visible Work-around MSR0 (OSVW\_ID\_Length)**

Reset: 0000 0000 0000 0000h.

Bits	Description
63:16	Reserved.
15:0	<b>OSVW_ID_Length: OS visible work-around ID length.</b> Read-write. See the <i>Revision Guide for AMD Family 10h Processors</i> for the definition of this field.

**MSRC001\_0141 OS Visible Work-around MSR1 (OSVW Status)**

Reset: 0000 0000 0000 0000h.

Bits	Description
63:0	<b>OswwStatusBits: OS visible work-around status bits.</b> Read-write. See the <i>Revision Guide for AMD Family 10h Processors</i> for the definition of this field.

**3.13 MSRs - MSRC001\_1xxx****MSRC001\_1004 CPUID Features Register (Features)**[MSRC001\\_1004](#) and [MSRC001\\_1005](#) provide some control over values read from CPUID functions.

Bits	Description
63:32	<b>Features.</b> Read-write. Provides back-door control over the features reported in CPUID function 1, ECX (see <a href="#">CPUID Fn0000_0001_ECX</a> ).
31:0	<b>Features.</b> Read-write. Provides back-door control over the features reported in CPUID function 1, EDX (see <a href="#">CPUID Fn[8000_0001, 0000_0001]_EDX</a> ).

**MSRC001\_1005 Extended CPUID Features Register (ExtFeatures)**See [MSRC001\\_1004](#).

Bits	Description
63:32	<b>ExtFeaturesEcx.</b> Read-write. Provides back-door control over the features reported in CPUID function 8000_0001, ECX (see <a href="#">CPUID Fn8000_0001_ECX</a> ). For G34, C32, S1g4, and ASB2 processors, BIOS should set ExtFeaturesEcx[19] with a read-modify-write.
31:0	<b>ExtFeaturesEdx.</b> Read-write. Provides back-door control over the features reported in CPUID function 8000_0001, EDX (see <a href="#">CPUID Fn[8000_0001, 0000_0001]_EDX</a> ).

**MSRC001\_100C Node ID**

Reset: 0000 0000 0000 0000h.

Bits	Description
63:12	Reserved.
11:6	<b>BiosScratch: bios scratch.</b> Read-write.
5:3	<b>NodesPerProcessor.</b> Read-write. Specifies the number of nodes per processor if <b>CPUID Fn8000_0001_ECX[NodeId]=1</b> . The number of nodes per processor is NodesPerProcessor+1. BIOS should program this to 001b if <b>F3xE8[MultiNodeCpu]=1</b> .
2:0	<b>NodeId.</b> Read-write. Specifies the node ID if <b>CPUID Fn8000_0001_ECX[NodeId]=1</b> . For G34 and C32 processors, BIOS should program this to the same value as <b>F0x60[NodeId]</b> .

### MSRC001\_1021 Instruction Cache Configuration Register (IC\_CFG)

Reset: 0000 0000 0000 0000h.

Bits	Description
63:15	Reserved.
14	<b>DIS_IND.</b> Read-write. 1=Disable indirect branch predictor. See 2.3.3 [Using L2 Cache as General Storage During Boot].
13:10	Reserved.
9	<b>DIS_SPEC_TLB_RLD.</b> Read-write. Disable speculative ITLB reloads. See 2.3.3 [Using L2 Cache as General Storage During Boot].
8:0	Reserved.

### MSRC001\_1022 Data Cache Configuration Register (DC\_CFG)

All defined fields are read-write.

Bits	Description
63:36	Reserved.
35:34	<b>REQ_CTR.</b> Reset: 11b. Initial number of requests (1 to 3) a hardware prefetch can make. Setting this field to 0 disables the hardware prefetcher. BIOS should program this to 01b for multi-link processors. A setting of 01b may also be preferred for single-link processors running server workloads.
33:14	Reserved.
13	<b>DIS_HW_PF.</b> Reset: 0. 1=Disable hardware prefetches. This is equivalent to setting REQ_CTR=0.
12:9	Reserved.
8	<b>DIS_CLR_WBTOL2_SMC_HIT.</b> Reset: 0. 1=Disable self modifying code checking logic. This is needed when using L2 cache as general storage. See 2.3.3 [Using L2 Cache as General Storage During Boot].
7:5	Reserved.
4	<b>DIS_SPEC_TLB_RLD.</b> Reset: 0. Disable speculative DTLB reloads. See 2.3.3 [Using L2 Cache as General Storage During Boot].
3:0	Reserved.

### MSRC001\_1023 Bus Unit Configuration Register (BU\_CFG)

Revision B2 and earlier revisions: Reset: 0000 0000 0000 0020h. Revision B3, C, and later revisions: Reset:

0000 0000 1020 0020h. All defined fields are read-write.

Bits	Description
63:49	Reserved.
48	<b>WbEnhWsbDis: disable multi-stream write combining.</b> 0=The bus unit performs write combining on up to 4 independent data streams. 1=The bus unit only performs write combining on a single data steam.
47:0	Reserved.

### **MSRC001\_102A Bus Unit Configuration 2 Register (BU\_CFG2)**

Revision B: Reset: 0000 0000 0100 80C0h. Revision C: Reset: 0000 0040 0100 80C0h. Revision D and later: Reset: 0000 0050 0100 8040h.

Bits	Description
63:51	Reserved.
50	<b>RdMmExtCfgQwEn: read mmio extended config quadword enable.</b> Revision B: Reserved. Revision C and later: Read-write. 1=MMIO reads to extended config space may be unaligned and up to quadword sized. 0=MMIO reads to extended config space must be doubleword aligned and doubleword size or smaller. BIOS should set this to 1.
42	<b>ProbeFilterSupEn.</b> Revisions C and earlier: Reserved. Revision D and later: Read-write. 1=Enable support for probe filter. See <a href="#">2.6.5.1 [Probe Filter]</a> .
37:36	<b>ThrottleNbInterface[3:2].</b> Revision C and earlier: Reserved. Revision D and later: Read-write. See <a href="#">ThrottleNbInterface[1:0]</a> .
29	<b>Smash1GPages.</b> Read-write. 1=1G pages are broken up and installed in the TLB as 2M pages.
28:16	Reserved.
15	<b>CILinesToNbDis.</b> Read-write. 1=Clean victims/copybacks are not sent to the NB. DC and IC caches WT/WP-IO, written to L2 when evicted but not written to L3 when evicted. 0=Clean victims/copybacks are implied to be from DRAM and are evicted from the IC/DC to the L2/L3. DC caches WT/WP-IO as NTA, which prevents the line from being written to the L2/L3 when evicted. IC does not cache WT/WP-IO. See <a href="#">2.3.3 [Using L2 Cache as General Storage During Boot]</a> . This bit has no effect if the L3 is disabled or not present.
14:8	Reserved.

7:6	<p><b>ThrottleNbInterface[1:0]</b>. Read-write. This field along with ThrottleNbInterface[3:2] specifies the number of clocks the core must wait between sending each packet of information to the NB. This field must be programmed to a value greater than or equal to the number of enabled cores minus 1. See <a href="#">2.9.2 [CPU Cores and Downcoring]</a>. BIOS should program this field to one less than the number of enabled cores in the node as follows:</p> <table border="1"> <thead> <tr> <th><u>Enabled Cores</u></th> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0000b</td> <td>0 Clocks</td> </tr> <tr> <td>2</td> <td>0001b</td> <td>1 Clock</td> </tr> <tr> <td>3</td> <td>0010b</td> <td>2 Clocks</td> </tr> <tr> <td>4</td> <td>0011b</td> <td>3 Clocks</td> </tr> <tr> <td colspan="3">Revision D only:</td> </tr> <tr> <td>5</td> <td>0100b</td> <td>4 Clocks</td> </tr> <tr> <td>6</td> <td>0101b</td> <td>5 Clocks</td> </tr> </tbody> </table> <p>All other settings are reserved.</p>	<u>Enabled Cores</u>	<u>Bits</u>	<u>Definition</u>	1	0000b	0 Clocks	2	0001b	1 Clock	3	0010b	2 Clocks	4	0011b	3 Clocks	Revision D only:			5	0100b	4 Clocks	6	0101b	5 Clocks
<u>Enabled Cores</u>	<u>Bits</u>	<u>Definition</u>																							
1	0000b	0 Clocks																							
2	0001b	1 Clock																							
3	0010b	2 Clocks																							
4	0011b	3 Clocks																							
Revision D only:																									
5	0100b	4 Clocks																							
6	0101b	5 Clocks																							
5:0	Reserved.																								

### MSRC001\_1030 IBS Fetch Control Register (IbsFetchCtl)

The IBS fetch sampling engine selects an instruction fetch to profile when the engine's periodic fetch counter reaches IbsFetchMaxCnt. The periodic fetch counter is an internal 20 bit counter that increments after every fetch cycle that completes when IbsFetchEn=1 and IbsFetchVal=0. When the selected instruction fetch completes or is aborted, the status of the fetch is written to the IBS fetch registers (this register, [MSRC001\\_1031](#) and [MSRC001\\_1032](#)) and an interrupt is generated. The interrupt service routine associated with this interrupt is responsible for saving the performance information stored in IBS fetch registers. See [2.16.2 \[Instruction Based Sampling \(IBS\)\]](#).

Bits	Description								
63:58	Reserved.								
57	<b>IbsRandEn: random instruction fetch tagging enable</b> . Read-write. Reset X. 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-only. Reset X. 1=The instruction fetch missed in the L2 TLB.								
55	<b>IbsL1TlbMiss: instruction cache L1TLB miss</b> . Read-only. Reset X. 1=The instruction fetch missed in the L1 TLB.								
54:53	<p><b>IbsL1TlbPgSz: instruction cache L1TLB page size</b>. Read-only. Reset X. This field indicates the page size of the translation in the L1 TLB. This field is only valid if IbsPhyAddrValid=1.</p> <table border="1"> <thead> <tr> <th><u>Bits</u></th> <th><u>Definition</u></th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>4 Kbyte</td> </tr> <tr> <td>01b</td> <td>2 Mbyte</td> </tr> <tr> <td>1Xb</td> <td>Reserved</td> </tr> </tbody> </table>	<u>Bits</u>	<u>Definition</u>	00b	4 Kbyte	01b	2 Mbyte	1Xb	Reserved
<u>Bits</u>	<u>Definition</u>								
00b	4 Kbyte								
01b	2 Mbyte								
1Xb	Reserved								
52	<b>IbsPhyAddrValid: instruction fetch physical address valid</b> . Read-only. Reset X. 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-only. Reset X. 1=The instruction fetch missed in the instruction cache.								
50	<b>IbsFetchComp: instruction fetch complete</b> . Read-only. 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. Reset 0b. 1=New instruction fetch data available. When this bit is set, the fetch counter stops counting and an interrupt is generated as specified by the APIC LVT specified by <a href="#">MSRC001_103A[LvtOffset]</a> . This bit must be cleared and <a href="#">IbsFetchCnt</a> must be written to 0000h for the fetch counter to start counting again.
48	<b>IbsFetchEn: instruction fetch enable.</b> Read-write. 1=Instruction fetch sampling is enabled.
47:32	<b>IbsFetchLat: instruction fetch latency.</b> Read-only. Reset X. This field indicates the number of clock cycles from when the instruction fetch was initiated to when the data was delivered to the core. If the instruction fetch is abandoned before the fetch completes, this field returns the number of clock cycles from when the instruction fetch was initiated to when the fetch was abandoned.
31:16	<b>IbsFetchCnt.</b> Read-write; updated-by-hardware. Reset 0000h. This field returns the current value of bits 19:4 of the periodic fetch counter.
15:0	<b>IbsFetchMaxCnt.</b> Read-write. Reset 0000h. This field specifies maximum count value of the periodic fetch counter. Programming this field to 0000h and setting <a href="#">IbsFetchEn</a> results in undefined behavior. Bits 19:4 of the maximum count are programmed in the field. Bits 3:0 of the maximum count are always 0000.

### MSRC001\_1031 IBS Fetch Linear Address Register ([IbsFetchLinAd](#))

Reset: xxxx xxxx xxxx xxxh.

Bits	Description
63:0	<b>IbsFetchLinAd: instruction fetch linear address.</b> Read-only. This field provides the linear address in canonical form for the tagged instruction fetch.

### MSRC001\_1032 IBS Fetch Physical Address Register ([IbsFetchPhysAd](#))

Reset: xxxx xxxx xxxx xxxh.

Bits	Description
63:0	<b>IbsFetchPhysAd: instruction fetch physical address.</b> Read-only. This 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[IbsPhyAddrValid]</a> is asserted.

### MSRC001\_1033 IBS Execution Control Register ([IbsOpCtl](#))

Reset: 0000 0000 0000 0000h. The IBS execution sampling engine tags a micro-op that will be issued in the next cycle to profile when the engine's periodic op counter reaches [IbsOpMaxCnt](#). The periodic op counter is an internal 20 bit counter that increments every cycle or dispatched op (see [IbsOpCntCtl](#)) when [IbsOpEn](#)=1 and [IbsOpVal](#)=0 and rolls over when the counter reaches [IbsOpMaxCnt](#). When the periodic op counter rolls over bits 6:0 of the counter are randomized by hardware. When the micro-op is retired, the status of the operation is written to the IBS execution registers (this register, [MSRC001\\_1034](#), [MSRC001\\_1035](#), [MSRC001\\_1036](#), [MSRC001\\_1037](#), [MSRC001\\_1038](#), [MSRC001\\_1039](#)) and an interrupt is generated. The interrupt service routine associated with this interrupt is responsible for saving the performance information stored in IBS execution registers. See [2.16.2 \[Instruction Based Sampling \(IBS\)\]](#).

Bits	Description
63:52	Reserved.

51:32	<b>IbsOpCurCnt: periodic op counter current count.</b> Revision B: Reserved. Revision C: Read-write. Reset X. This field returns the current value of the periodic op counter.
31:20	Reserved.
19	<b>IbsOpCntCtl: periodic op counter count control.</b> Revision B: Reserved. Revision C: Read-write. Reset 0b. 1=Count dispatched ops 0=Count clock cycles.
18	<b>IbsOpVal: micro-op sample valid.</b> Read-write; set-by-hardware. 1=New instruction execution data available. When this bit is set, the periodic op counter stops counting until software clears the bit. An interrupt may be generated 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: periodic op counter maximum count.</b> Read-write. Reset X. This field specifies maximum count value of the periodic op counter. Bits 19:4 of the maximum count are programmed in the field. Bits 3:0 of the maximum count are always 0000.

### MSRC001\_1034 IBS Op Logical Address Register (IbsOpRip)

Reset: xxxx xxxx xxxx xxxh.

Bits	Description
63:0	<b>IbsOpRip: micro-op linear address.</b> Read-write. Linear address in canonical form for the instruction that contains the tagged micro-op.

### MSRC001\_1035 IBS Op Data Register (IbsOpData)

Bits	Description
63:38	Reserved.
37	<b>IbsOpBrnRet: branch micro-op retired.</b> Reset X. Read-only. 1=Tagged operation was a branch that retired.
36	<b>IbsOpBrnMisp: mispredicted branch micro-op.</b> Read-only. Reset X. 1=Tagged operation was a branch micro-op that was mispredicted.
35	<b>IbsOpBrnTaken: taken branch micro-op.</b> Read-only. Reset X. 1=Tagged operation was a branch micro-op that was taken.
34	<b>IbsOpReturn: return micro-op.</b> Read-only. Reset X. 1=Tagged operation was return micro-op.
33	<b>IbsOpMispReturn: mispredicted return micro-op.</b> Read-only. Reset X. 1=Tagged operation was a mispredicted return micro-op.
32	<b>IbsOpBrnResync: resync micro-op.</b> Read-only. Reset X. 1=Tagged operation was resync micro-op.
31:16	<b>IbsTagToRetCtr: micro-op tag to retire count.</b> Read-only. Reset X. 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 <a href="#">IbsCompToRetCtr</a> when the tagged micro-op is a NOP.
15:0	<b>IbsCompToRetCtr: micro-op completion to retire count.</b> Read-only. Reset X. This field returns the number of cycles from when the micro-op was completed to when the micro-op was retired.

**MSRC001\_1036 IBS Op Data 2 Register (IbsOpData2)**

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.

63:6	Reserved.
5	<b>NbIbsReqCacheHitSt: IBS L3 cache state.</b> Read-write; updated-by-hardware. Valid when the data source type is Cache (NbIbsReqSrc=2h). 0 = 'M' State. 1 = 'O' State.
4	<b>NbIbsReqDstProc: IBS request destination processor.</b> Read-write; updated-by-hardware. 0=The request is serviced by the NB in the same node as the core. 1=The request is serviced by the NB in a different node than the core. Valid when NbIbsReqSrc is non-zero.
3	Reserved.
2:0	<b>NbIbsReqSrc: Northbridge IBS request data source.</b> Read-write. 0h=No valid status. 4h=Reserved for remote cache. 1h=L3: data returned from local L3 cache. 5h=Reserved. 2h=Cache: data returned from a CPU cache or a remote L3. 6h=Reserved. 3h=DRAM: data returned from DRAM. 7h=Other: data returned from MMIO/Config/PCI/APIC.

**MSRC001\_1037 IBS Op Data 3 Register (IbsOpData3)**

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.

63:48	Reserved.
47:32	<b>IbsDcMissLat: data cache miss latency.</b> Read-only. Reset X. This field indicates the number of clock cycles from when a miss is detected in the data cache to when the data was delivered to the core. The value returned by this counter is not valid for data cache writes or prefetch instructions.
31:20	Reserved
19	<b>IbsDcL2tlbHit1G: data cache L2TLB hit in 1G page.</b> Revision B: Reserved. Revision C: Read-only. Reset X. 1=The physical address for the tagged load or store operation was present in a 1G page table entry in the data cache L2TLB.
18	<b>IbsDcPhyAddrValid: data cache physical address valid.</b> Read-only. Reset X. 1=The physical address in <a href="#">MSRC001_1039</a> is valid for the load or store operation.
17	<b>IbsDcLinAddrValid: data cache linear address valid.</b> Read-only. Reset X. 1=The linear address in <a href="#">MSRC001_1038</a> is valid for the load or store operation.
16	<b>IbsDcMabHit: MAB hit.</b> Read-only. Reset X. 1=The tagged load or store operation hit on an already allocated MAB.
15	<b>IbsDcLockedOp: locked operation.</b> Read-only. Reset X. 1=Tagged load or store operation is a locked operation.
14	<b>IbsDcUcMemAcc: UC memory access.</b> Read-only. Reset X. 1=Tagged load or store operation accessed uncacheable memory.
13	<b>IbsDcWcMemAcc: WC memory access.</b> Read-only. Reset X. 1=Tagged load or store operation accessed write combining memory.
12	<b>IbsDcStToLdCan: data forwarding from store to load operation cancelled.</b> Read-only. Reset X. 1=Data forwarding from a store operation to the tagged load was cancelled.

11	<b>IbsDcStToLdFwd: data forwarded from store to load operation.</b> Read-only. Reset X. 1=Data for tagged load operation was forwarded from a store operation. If this bit is set and IbsDcStToLdCan=1, then the data for the load operation forwarded from a store operation but the data was not forwarded immediately.
10	<b>IbsDcStBnkCon: bank conflict on store operation.</b> Read-only. Reset X. 1=A bank conflict with a store operation occurred in the data cache on the tagged load or store operation.
9	<b>IbsDcLdBnkCon: bank conflict on load operation.</b> Read-only. Reset X. 1=A bank conflict with a load operation occurred in the data cache on the tagged load or store operation.
8	<b>IbsDcMisAcc: misaligned access.</b> Read-only. Reset X. 1=The tagged load or store operation crosses a 128 bit address boundary.
7	<b>IbsDcMiss: data cache miss.</b> Read-only. Reset X. 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-only. Reset X. 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	<b>IbsDcL1tlbHit1G: data cache L1TLB hit in 1G page.</b> Read-only. Reset X. 1=The physical address for the tagged load or store operation was present in a 1G page table entry in the data cache L1TLB.
4	<b>IbsDcL1tlbHit2M: data cache L1TLB hit in 2M page.</b> Read-only. Reset X. 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-only. Reset X. 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-only. Reset X. 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-only. Reset X. 1=Tagged operation is a store operation
0	<b>IbsLdOp: load op.</b> Read-only. Reset X. 1=Tagged operation is a load operation

### MSRC001\_1038 IBS DC Linear Address Register (IbsDcLinAd)

Reset: xxxx xxxx xxxx xxxh.

Bits	Description
63:0	<b>IbsDcLinAd.</b> Read-only. This field 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 Register (IbsDcPhysAd)

Reset: xxxx xxxx xxxx xxxh.

Bits	Description
63:0	<b>IbsDcPhysAd.</b> Read-only. This 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 Register

Reset: 0000 0000 0000 0000h. This register is a read-only copy of F3x1CC.

Bits	Description
63:9	Reserved.
8	<b>LvtOffsetVal: local vector table offset valid.</b> Read-only. 1=The offset in LvtOffset is valid.
7:4	Reserved.
3:0	<b>LvtOffset: local vector table offset.</b> Read-only. This specifies the address of the IBS LVT entry in the APIC registers as follows: LVT address = (LvtOffset << 4) + 500h (see <a href="#">APIC[530:500]</a> ).

### 3.14 Performance Counter Events

This section provides the performance counter events that may be selected through [\[The Performance Event Select Register \(PERF\\_CTL\[3:0\]\)\] MSRC001\\_00\[03:00\]\[EventSelect and UnitMask\]](#). See that register and [\[The Performance Event Counter Registers \(PERF\\_CTR\[3:0\]\)\] MSRC001\\_00\[07:04\]](#) for details.

#### 3.14.1 Floating Point Events

See the following events for additional floating point information:

- [EventSelect 0CBh \[Retired MMX™/FP Instructions\]](#).
- [EventSelect 0DBh \[FPU Exceptions\]](#).
- [EventSelect 1C0h \[Retired x87 Floating Point Operations\]](#).

#### **EventSelect 000h Dispatched FPU Operations**

The number of operations (uops) dispatched to the FPU execution pipelines. This event reflects how busy the FPU pipelines are. This includes all operations done by x87, MMX™ and SSE instructions, including moves. Each increment represents a one-cycle dispatch event; packed 128-bit SSE operations count as two ops in 64-bit FPU implementations; scalar operations count as one. This event is a speculative event. (See [EventSelect 0CBh](#)). Note: Since this event includes non-numeric operations it is not suitable for measuring MFLOPs.

UnitMask	Description
01h	Add pipe ops excluding load ops and SSE move ops
02h	Multiply pipe ops excluding load ops and SSE move ops
04h	Store pipe ops excluding load ops and SSE move ops
08h	Add pipe load ops and SSE move ops
10h	Multiply pipe load ops and SSE move ops
20h	Store pipe load ops and SSE move ops

#### **EventSelect 001h Cycles in which the FPU is Empty**

The number of cycles in which the FPU is empty. Invert this ([MSRC001\\_00\[03:00\]\[Invert\]=1](#)) to count cycles in which at least one FPU operation is present in the FPU.

#### **EventSelect 002h 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.

### EventSelect 003h Retired SSE Operations

The number of SSE operations retired. This counter can count either FLOPS (UnitMask bit 6 = 1) or uops (UnitMask bit 6 = 0).

UnitMask	Description
01h	Single precision add/subtract ops
02h	Single precision multiply ops
04h	Single precision divide/square root ops
08h	Double precision add/subtract ops
10h	Double precision multiply ops
20h	Double precision divide/square root ops
40h	Op type: 0=uops. 1=FLOPS

### EventSelect 004h Retired Move Ops

The number of move uops retired. Merging low quadword move ops copy the lower 64 bits of a source register to the upper 64 bits of a destination register. The lower 64 bits of the destination register remain unchanged. Merging high quadword move ops copy the upper 64 bits of a source register to the lower 64 bits of a destination register. The upper 64 bits of the destination register remain unchanged.

UnitMask	Description
01h	Merging low quadword move uops
02h	Merging high quadword move uops
04h	All other merging move uops
08h	All other move uops

### EventSelect 005h Retired Serializing Ops

The number of serializing uops retired. A bottom-executing uop is not issued until it is the oldest non-retired uop in the FPU. Bottom-executing ops are most commonly seen with FSTSW and STMXCSR instructions. A bottom-serializing uop does not issue until it is the oldest non-issued uop in the FP scheduler. Bottom-serializing uops block all subsequent uops from being issued until the uop is issued. Bottom-serializing ops are most commonly seen with FLCDW and LDMXCSR instructions.

UnitMask	Description
01h	SSE bottom-executing uops retired
02h	SSE bottom-serializing uops retired
04h	x87 bottom-executing uops retired
08h	x87 bottom-serializing uops retired

### EventSelect 006h Number of Cycles that a Serializing uop is in the FP Scheduler

See [EventSelect 005h](#) for a description of bottom-executing and bottom-serializing uop.

UnitMask	Description
01h	Number of cycles a bottom-execute uop is in the FP scheduler
02h	Number of cycles a bottom-serializing uop is in the FP scheduler

### 3.14.2 Load/Store and TLB Events

See the following events for additional Load/Store and TLB information:

- [EventSelect 065h \[Memory Requests by Type\]](#).

#### **EventSelect 020h Segment Register Loads**

The number of segment register loads performed.

UnitMask	Description
01h	ES
02h	CS
04h	SS
08h	DS
10h	FS
20h	GS
40h	HS

#### **EventSelect 021h Pipeline Restart Due to Self-Modifying Code**

The number of pipeline restarts caused by self-modifying code (a store that hits any instruction that has been fetched for execution beyond the instruction doing the store).

#### **EventSelect 022h Pipeline Restart Due to Probe Hit**

The number of pipeline restarts caused by an invalidating probe hitting on a speculative out-of-order load.

#### **EventSelect 023h LS Buffer 2 Full**

The number of cycles that the LS2 buffer is full. This buffer holds stores waiting to retire as well as requests that missed the data cache and are waiting on a refill. This condition stalls further data cache accesses, although such stalls may be overlapped by independent instruction execution.

#### **EventSelect 024h 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
01h	The number of locked instructions executed
02h	The number of cycles spent in speculative phase
04h	The number of cycles spent in non-speculative phase (including cache miss penalty)
08h	The number of cycles waiting for a cache hit (cache miss penalty).



**EventSelect 026h Retired CLFLUSH Instructions**

The number of CLFLUSH instructions retired.

**EventSelect 027h Retired CUID Instructions**

The number of CUID instructions retired.

**EventSelect 02Ah Cancelled Store to Load Forward Operations**

Counts the number store to load forward operations that are cancelled.

UnitMask	Description
01h	Address mismatches (starting byte not the same).
02h	Store is smaller than load.
04h	Misaligned.

**EventSelect 02Bh SMIs Received**

Counts the number of SMIs received by the processor.

**3.14.3 Data Cache Events****EventSelect 040h 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.

**EventSelect 041h Data Cache Misses**

The number of data cache references which missed in the data cache. This event is a speculative event.

Except in the case of streaming stores, only the first miss for a given line is included - access attempts by other instructions while the refill is still pending are not included in this event. So in the absence of streaming stores, each event reflects one 64-byte cache line refill, and counts of this event are the same as, or very close to, the combined count for [EventSelect 042h](#).

Streaming stores however cause this event for every such store, since the target memory is not refilled into the cache. Hence this event should not be used as an indication of data cache refill activity - [EventSelect 042h](#) should be used for such measurements. See [EventSelect 065h](#) for an indication of streaming store activity. A large difference between this event (with all UnitMask bits set) and [EventSelect 042h](#) would be due mainly to streaming store activity and hardware prefetch requests.

For revision C and later revisions, this event no longer counts data cache misses associated with streaming stores.

**EventSelect 042h 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. UnitMask bits 4:1 allow a breakdown of refills from the L2 by coherency state. UnitMask bit 0 reflects refills

which missed in the L2, and provides the same measure as the combined sub-events of [EventSelect 043h](#). Each increment reflects a 64-byte transfer. This event is a speculative event.

UnitMask	Description
01h	Refill from the Northbridge
02h	Shared-state line from L2
04h	Exclusive-state line from L2
08h	Owned-state line from L2
10h	Modified-state line from L2

### **EventSelect 043h Data Cache Refills from the Northbridge**

The number of L1 cache refills satisfied from the Northbridge (DRAM, L3 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
01h	Invalid
02h	Shared
04h	Exclusive
08h	Owned
10h	Modified

### **EventSelect 044h Data Cache Lines Evicted**

The number of L1 data cache lines written to the L2 cache or system memory, having been displaced by L1 refills. The UnitMask may be used to count only victims in specific coherency states. Each increment represents a 64-byte transfer. This event is a speculative event.

In most cases, L1 victims are moved to the L2 cache, displacing an older cache line there. Lines brought into the data cache by PrefetchNTA instructions, however, are evicted directly to system memory (if dirty) or invalidated (if clean). The Invalid case (UnitMask value 01h) reflects the replacement of lines that would have been invalidated by probes for write operations from another processor or DMA activity. UnitMask 20h and 40h count all evictions regardless of cache line state. When either UnitMask 20h or 40h is enabled all other UnitMasks should be disabled.

UnitMask	Description
01h	Invalid
02h	Shared
04h	Exclusive
08h	Owned
10h	Modified
20h	Cache line evicted was brought into the cache with by a PrefetchNTA instruction.
40h	Cache line evicted was not brought into the cache with by a PrefetchNTA instruction.

**EventSelect 045h L1 DTLB Miss and L2 DTLB Hit**

The number of data cache accesses that miss in the L1 DTLB and hit in the L2 DTLB. This event is a speculative event.

UnitMask	Description
01h	L2 4K TLB hit
02h	L2 2M TLB hit
04h	Revision B: Reserved. Revision C: L2 1G TLB hit

**EventSelect 046h L1 DTLB and L2 DTLB Miss**

The number of data cache accesses that miss in both the L1 and L2 DTLBs. This event is a speculative event.

UnitMask	Description
01h	4K TLB reload
02h	2M TLB reload
04h	1G TLB reload

**EventSelect 047h Misaligned Accesses**

The number of data cache accesses that are misaligned. These are accesses which cross a sixteen-byte boundary. They incur an extra cache access (reflected in [EventSelect 040h](#)), and an extra cycle of latency on reads. This event is a speculative event.

**EventSelect 048h Microarchitectural Late Cancel of an Access****EventSelect 049h Microarchitectural Early Cancel of an Access****EventSelect 04Ah Single-bit ECC Errors Recorded by Scrubber**

The number of single-bit errors corrected by either of the error detection/correction mechanisms in the data cache.

UnitMask	Description
01h	Scrubber error
02h	Piggyback scrubber errors
04h	Load pipe error
08h	Store write pipe error

**EventSelect 04Bh Prefetch Instructions Dispatched**

The number of prefetch instructions dispatched by the decoder. Such instructions may or may not cause a cache line transfer. All Dcache and L2 accesses, hits and misses by prefetch instructions, except for prefetch instructions that collide with an outstanding hardware prefetch, are included in these events. This event is a speculative event.

UnitMask	Description
01h	Load (Prefetch, PrefetchT0/T1/T2)
02h	Store (PrefetchW)
04h	NTA (PrefetchNTA)

### **EventSelect 04Ch DCACHE Misses by Locked Instructions**

The number of data cache misses incurred by locked instructions. (The total number of locked instructions may be obtained from [EventSelect 024h](#).)

Such misses may be satisfied from the L2 or system memory, but there is no provision for distinguishing between the two. When used for event-based profiling, this event tends to occur very close to the offending instructions. This event is also included in the basic Dcache miss event ([EventSelect 041h](#)).

UnitMask	Description
02h	Data cache misses by locked instructions

### **EventSelect 04Dh L1 DTLB Hit**

The number of data cache accesses that hit in the L1 DTLB. This event is a speculative event.

UnitMask	Description
01h	L1 4K TLB hit
02h	L1 2M TLB hit
04h	L1 1G TLB hit

### **EventSelect 052h Ineffective Software Prefetchs**

The number of software prefetches that did not fetch data outside of the processor core.

UnitMask	Description
01h	Software prefetch hit in the L1.
08h	Software prefetch hit in L2.

### **EventSelect 054h Global TLB Flushes**

This event counts TLB flushes that flush TLB entries that have the global bit set.

#### **3.14.4 L2 Cache and System Interface Events**

### **EventSelect 065h Memory Requests by Type**

These events reflect accesses to uncacheable (UC) or write-combining (WC) memory regions (as defined by MTRR or PAT settings) and Streaming Store activity to WB memory. Both the WC and Streaming Store events reflect Write Combining buffer flushes, not individual store instructions. WC buffer flushes which typically consist of one 64-byte write to the system for each flush (assuming software typically fills a buffer before it gets flushed). A partially-filled buffer requires two or more smaller writes to the system. The WC event reflects flushes of WC buffers that are filled by stores to WC memory or streaming stores to WB memory. The Stream-

ing Store event reflects only flushes due to streaming stores (which are typically only to WB memory). The difference between counts of these two events reflects the true amount of write events to WC memory.

UnitMask	Description
01h	Requests to non-cacheable (UC) memory
02h	Requests to write-combining (WC) memory or WC buffer flushes to WB memory
80h	Streaming store (SS) requests

### EventSelect 067h Data Prefetcher

These events reflect requests made by the data prefetcher. UnitMask bit 1 counts total prefetch requests, while bit 0 counts requests where the target block is found in the L2 or data cache. The difference between the two represents actual data read (in units of 64-byte cache lines) from the system by the prefetcher. This is also included in the count of [EventSelect 07Fh](#), UnitMask bit 0 (combined with other L2 fill events). This event is always 0 because the processor prefetches into the data cache, not the L2 cache.

UnitMask	Description
01h	Cancelled prefetches
02h	Prefetch attempts

### EventSelect 068h MAB Requests

#### EventSelect 069h MAB Wait Cycles

These two event selects provide a measure of the average L1 refill latency for Icache and Dcache misses. [EventSelect 068h](#) provides a partial request count for cache refills, and [EventSelect 069h](#) provides a count of cycles those requests spent waiting for the refills to complete. Dividing the cycle count by a request count that was measured over the same interval with the same UnitMask gives the average refill latency in processor clock cycles. The refill times include cases that are satisfied from higher cache levels as well as from system DRAM. It is not meaningful to combine the UnitMask values.

The UnitMask is an encoded value which selects one of the ten Miss Address Buffers (MABs). Buffers 0-7 handle Dcache misses, while buffers 8 and 9 handle Icache misses. MAB entries 1 and 8 give 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
N	Buffer N, where N = 0-9

### EventSelect 06Ch Northbridge Read Responses by Coherency State

The number of responses from the Northbridge for cache refill requests. The UnitMask may be used to select specific cache coherency states. Each increment represents one 64-byte cache line transferred from the Northbridge (DRAM, L3, 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
01h	Exclusive
02h	Modified
04h	Shared
08h	Owned
10h	Data Error

### **EventSelect 06Dh Octwords Written to System**

The number of octword (16-byte) data transfers from the processor to the system. These may be part of a 64-byte cache line writeback or a 64-byte dirty probe hit response, each of which would cause four increments; or a partial or complete Write Combining buffer flush (Sized Write), which could cause from one to four increments.

UnitMask	Description
01h	Octword write transfer

### **EventSelect 076h 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.

### **EventSelect 07Dh Requests to L2 Cache**

The number of requests to the L2 cache for Icache or Dcache fills, or page table lookups for the TLB. These events reflect only read requests to the L2. These include some amount of retries associated with address or resource conflicts. Such retries tend to occur more as the L2 gets busier, and in certain extreme cases (such as large block moves that overflow the L2) these extra requests can dominate the event count.

These extra requests are not a direct indication of performance impact - they simply reflect opportunistic accesses that don't complete. But because of this, they are not a good indication of actual cache line movement. The Icache and Dcache miss and refill events (81h, 82h, 83h, 41h, 42h, 43h) provide a more accurate indication of this, and are the preferred way to measure such traffic.

UnitMask	Description
01h	IC fill
02h	DC fill
04h	TLB fill (page table walks)
08h	Tag snoop request
10h	Cancelled request
20h	Hardware prefetch from DC

### **EventSelect 07Eh L2 Cache Misses**

The number of requests that miss in the L2 cache. This may include some amount of speculative activity, as well as some amount of retried requests as described in [EventSelect 07Dh](#). The IC-fill-miss and DC-fill-miss events tend to mirror the Icache and Dcache refill-from-system events (83h and [EventSelect 043h](#), respectively), and tend to include more speculative activity than those events.

UnitMask	Description
01h	IC fill
02h	DC fill (includes possible replays, whereas <a href="#">EventSelect 041h</a> does not)
04h	TLB page table walk
08h	Hardware prefetch from DC

### **EventSelect 07Fh L2 Fill/Writeback**

The number of lines written into the L2 cache due to victim writebacks from the Icache or Dcache, TLB page table walks and the hardware data prefetcher (UnitMask bit 0); or writebacks of dirty lines from the L2 to the system (UnitMask bit 1). Each increment represents a 64-byte cache line transfer.

Note: Victim writebacks from the Dcache may be measured separately using [EventSelect 044h](#). However this is not quite the same as the Dcache component of this event, the main difference being PrefetchNTA lines. When these are evicted from the Dcache due to replacement, they are written out to system memory (if dirty) or simply invalidated (if clean), rather than being moved to the L2 cache.

UnitMask	Description
01h	L2 fills (victims from L1 caches, TLB page table walks and data prefetches)
02h	L2 Writebacks to system.

### **EventSelect 165h Page Size Mismatches**

Revision B: Reserved. Revision C and later revisions: Counts the number of large pages that are installed into the TLB as a smaller page size when nested paging is enabled.

UnitMask	Description
01h	Guest page size is larger than the host page size.
02h	MTRR mismatch.
04h	Host page size is larger than the guest page size.

### **3.14.5 Instruction Cache Events**

Note: All instruction cache events are speculative events unless specified otherwise.

#### **EventSelect 080h Instruction Cache Fetches**

The number of instruction cache accesses by the instruction fetcher. Each access is an aligned 16 byte read, from which a varying number of instructions may be decoded.

#### **EventSelect 081h 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.



**EventSelect 082h 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.

**EventSelect 083h 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.

**EventSelect 084h L1 ITLB Miss, L2 ITLB Hit**

The number of instruction fetches that miss in the L1 ITLB but hit in the L2 ITLB.

**EventSelect 085h L1 ITLB Miss, L2 ITLB Miss**

The number of instruction fetches that miss in both the L1 and L2 ITLBs.

UnitMask	Description
01h	Instruction fetches to a 4K page.
02h	Instruction fetches to a 2M page.

**EventSelect 086h Pipeline Restart Due to Instruction Stream Probe**

The number of pipeline restarts caused by invalidating probes that hit on the instruction stream currently being executed. This would happen if the active instruction stream was being modified by another processor in an MP system - typically a highly unlikely event.

**EventSelect 087h 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.

**EventSelect 088h 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.

Note: 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.

**EventSelect 089h 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.

**EventSelect 08Bh Instruction Cache Victims**

The number of cachelines evicted from the instruction cache to the L2.

**EventSelect 08Ch Instruction Cache Lines Invalidated**

---

The number of instruction cache lines invalidated.

UnitMask	Description
01h	Invalidating probe that did not hit any in-flight instructions.
02h	Invalidating probe that hit one or more in-flight instructions.

**EventSelect 099h ITLB Reloads**

---

The number of ITLB reload requests.

**EventSelect 09Ah ITLB Reloads Aborted**

---

The number of ITLB reloads aborted.

**3.14.6 Execution Unit Events**

See the following events for additional execution unit information:

- [EventSelect 026h \[Retired CLFLUSH Instructions\]](#).
- [EventSelect 027h \[Retired CPUID Instructions\]](#).
- [EventSelect 076h \[CPU Clocks not Halted\]](#).

**EventSelect 0C0h 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.

**EventSelect 0C1h Retired uops**

---

The number of micro-ops retired. This includes all processor activity (instructions, exceptions, interrupts, microcode assists, etc.).

**EventSelect 0C2h Retired Branch Instructions**

---

The number of branch instructions retired. This includes all types of architectural control flow changes, including exceptions and interrupts.

**EventSelect 0C3h Retired Mispredicted Branch Instructions**

---

The number of branch instructions retired, of any type, that are not correctly predicted. This includes those for which prediction is not attempted (far control transfers, exceptions and interrupts).

**EventSelect 0C4h Retired Taken Branch Instructions**

---

The number of taken branches retired. This includes all types of architectural control flow changes, including exceptions and interrupts.

**EventSelect 0C5h Retired Taken Branch Instructions Mispredicted**

---

The number of retired taken branch instructions that are mispredicted.

**EventSelect 0C6h Retired Far Control Transfers**

---

The number of far control transfers retired including far call/jump/return, IRET, SYSCALL and SYSRET, plus

exceptions and interrupts. Far control transfers are not subject to branch prediction.

#### **EventSelect 0C7h 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.

#### **EventSelect 0C8h Retired Near Returns**

The number of near return instructions (RET or RET Iw) retired.

#### **EventSelect 0C9h Retired Near Returns Mispredicted**

The number of near returns retired that are not correctly predicted by the return address predictor. Each such mispredict incurs the same penalty as a mispredicted conditional branch instruction.

#### **EventSelect 0CAh Retired Indirect Branches Mispredicted**

The number of indirect branch instructions retired where the target address was not correctly predicted.

#### **EventSelect 0CBh Retired MMX™/FP Instructions**

The number of MMX™, SSE or X87 instructions retired. The UnitMask allows the selection of the individual classes of instructions as given in the table. Each increment represents one complete instruction.

Note: Since this event includes non-numeric instructions it is not suitable for measuring MFLOPS.

UnitMask	Description
01h	x87 instructions
02h	MMX™ and 3DNow!™ instructions
04h	SSE instructions (SSE, SSE2, SSE3, and SSE4A)

#### **EventSelect 0CCh Retired Fastpath Double Op Instructions**

UnitMask	Description
01h	With low op in position 0
02h	With low op in position 1
04h	With low op in position 2

#### **EventSelect 0CDh 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 [EventSelect 0CFh](#) to determine how often interrupts are disabled for interrupt handling vs. other reasons (e.g. critical sections).

#### **EventSelect 0CEh 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 [EventSelect 0CDh](#) 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.

---

**EventSelect 0CFh Interrupts Taken**

---

The number of hardware interrupts taken. This does not include software interrupts (INT n instruction).

---

**EventSelect 0D0h Decoder Empty**

---

The number of processor cycles where the decoder has nothing to dispatch (typically waiting on an instruction fetch that missed the Icache, or for the target fetch after a branch mispredict).

---

**EventSelect 0D1h Dispatch Stalls**

---

The number of processor cycles where the decoder is stalled for any reason (has one or more instructions ready but can't dispatch them due to resource limitations in execution). This is the combined effect of events D2h - DAh, some of which may overlap; this event reflects the net stall cycles. The more common stall conditions (events D5h, D6h, D7h, D8h, and to a lesser extent D2) may overlap considerably. The occurrence of these stalls is highly dependent on the nature of the code being executed (instruction mix, memory reference patterns, etc.).

---

**EventSelect 0D2h Dispatch Stall for Branch Abort to Retire**

---

The number of processor cycles the decoder is stalled waiting for the pipe to drain after a mispredicted branch. This stall occurs if the corrected target instruction reaches the dispatch stage before the pipe has emptied. See [EventSelect 0D1h](#).

---

**EventSelect 0D3h Dispatch Stall for Serialization**

---

The number of processor cycles the decoder is stalled due to a serializing operation, which waits for the execution pipeline to drain. Relatively rare; mainly associated with system instructions. See [EventSelect 0D1h](#).

---

**EventSelect 0D4h Dispatch Stall for Segment Load**

---

The number of processor cycles the decoder is stalled due to a segment load instruction being encountered while execution of a previous segment load operation is still pending. Relatively rare except in 16-bit code. See [EventSelect 0D1h](#).

---

**EventSelect 0D5h Dispatch Stall for Reorder Buffer Full**

---

The number of processor cycles the decoder is stalled because the reorder buffer is full. May occur simultaneously with certain other stall conditions; see [EventSelect 0D1h](#).

---

**EventSelect 0D6h Dispatch Stall for Reservation Station Full**

---

The number of processor cycles the decoder is stalled because a required integer unit reservation stations is full. May occur simultaneously with certain other stall conditions; see [EventSelect 0D1h](#).

---

**EventSelect 0D7h Dispatch Stall for FPU Full**

---

The number of processor cycles the decoder is stalled because the scheduler for the Floating Point Unit is full. This condition can be caused by a lack of parallelism in FP-intensive code, or by cache misses on FP operand loads (which could also show up as [EventSelect 0D8h](#) instead, depending on the nature of the instruction sequences). May occur simultaneously with certain other stall conditions; see [EventSelect 0D1h](#).

---

**EventSelect 0D8h Dispatch Stall for LS Full**

---

The number of processor cycles the decoder is stalled because the Load/Store Unit is full. This generally occurs due to heavy cache miss activity. May occur simultaneously with certain other stall conditions; see [EventSelect 0D1h](#).

**EventSelect 0D9h Dispatch Stall Waiting for All Quiet**

The number of processor cycles the decoder is stalled waiting for all outstanding requests to the system to be resolved. Relatively rare; associated with certain system instructions and types of interrupts. May partially overlap certain other stall conditions; see [EventSelect 0D1h](#).

**EventSelect 0DAh Dispatch Stall for Far Transfer or Resync to Retire**

The number of processor cycles the decoder is stalled waiting for the execution pipeline to drain before dispatching the target instructions of a far control transfer or a Resync (an instruction stream restart associated with certain microcode assists). Relatively rare; does not overlap with other stall conditions. See [EventSelect 0D1h](#).

**EventSelect 0DBh 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
01h	x87 reclass microfaults
02h	SSE retype microfaults
04h	SSE reclass microfaults
08h	SSE and x87 microtraps

**EventSelect 0DCh DR0 Breakpoint Matches**

The number of matches on the address in breakpoint register DR0, per the breakpoint type specified in DR7. The breakpoint does not have to be enabled. Each instruction breakpoint match incurs an overhead of about 120 cycles; load/store breakpoint matches do not incur any overhead.

**EventSelect 0DDh DR1 Breakpoint Matches**

The number of matches on the address in breakpoint register DR1. See notes for [EventSelect 0DCh](#).

**EventSelect 0DEh DR2 Breakpoint Matches**

The number of matches on the address in breakpoint register DR2. See notes for [EventSelect 0DCh](#).

**EventSelect 0DFh DR3 Breakpoint Matches**

The number of matches on the address in breakpoint register DR3. See notes for [EventSelect 0DCh](#).

**EventSelect 1C0h Retired x87 Floating Point Operations**

Revision B: Reserved. Revision C and later revisions: The number of x87 floating point ops that have retired.

UnitMask	Description
01h	Add/subtract ops
02h	Multiply ops
04h	Divide ops

**EventSelect 1CFh IBS Ops Tagged**

Revision B: Reserved. Revision C and later revisions: The number of ops tagged by IBS.

#### **EventSelect 1D3h LFENCE Instructions Retired**

Revision B: Reserved. Revision C and later revisions: The number of LFENCE instructions retired.

#### **EventSelect 1D4h SFENCE Instructions Retired**

Revision B: Reserved. Revision C and later revisions: The number of SFENCE instructions retired. This counter only counts properly if `MSRC001_0015[SlowSfence]=0`.

#### **EventSelect 1D5h MFENCE Instructions Retired**

Revision B: Reserved. Revision C and later revisions: The number of MFENCE instructions retired.

### **3.14.7 Memory Controller Events**

#### **EventSelect 0E0h DRAM Accesses**

The number of memory accesses performed by the local DRAM controller. The UnitMask may be used to isolate the different DRAM page access cases. Page miss cases incur an extra latency to open a page; page conflict cases incur both a page-close as well as page-open penalties. These penalties may be overlapped by DRAM accesses for other requests and don't necessarily represent lost DRAM bandwidth. The associated penalties are as follows:

Page miss: Trcd (DRAM RAS-to-CAS delay)  
 Page conflict: Trp + Trcd (DRAM row-precharge time plus RAS-to-CAS delay)

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
01h	DCT0 Page hit
02h	DCT0 Page Miss
04h	DCT0 Page Conflict
08h	DCT1 Page hit
10h	DCT1 Page Miss
20h	DCT1 Page Conflict

#### **EventSelect 0E1h 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
01h	DCT0 Page Table Overflow
02h	DCT1 Page Table Overflow

**EventSelect 0E2h Memory Controller DRAM Command Slots Missed**

UnitMask	Description
01h	DCT0 Command Slots Missed
02h	DCT1 Command Slots Missed

**EventSelect 0E3h Memory Controller Turnarounds**

The number of turnarounds on the local DRAM data bus. The UnitMask may be used to isolate the different cases. These represent lost DRAM bandwidth, which may be calculated as follows (in bytes per occurrence):

DIMM turnaround:  $\text{DRAM\_width\_in\_bytes} * 2 \text{ edges\_per\_memclk} * 2$   
 R/W turnaround:  $\text{DRAM\_width\_in\_bytes} * 2 \text{ edges\_per\_memclk} * 1$   
 R/W turnaround:  $\text{DRAM\_width\_in\_bytes} * 2 \text{ edges\_per\_memclk} * (\text{Tcl}-1)$

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
01h	DCT0 DIMM (chip select) turnaround
02h	DCT0 Read to write turnaround
04h	DCT0 Write to read turnaround
08h	DCT1 DIMM (chip select) turnaround
10h	DCT1 Read to write turnaround
20h	DCT1 Write to read turnaround

**EventSelect 0E4h Memory Controller Bypass Counter Saturation**

UnitMask	Description
01h	Memory controller high priority bypass
02h	Memory controller medium priority bypass
04h	DCT0 DCQ bypass
08h	DCT1 DCQ bypass

**EventSelect 0E8h Thermal Status**

UnitMask	Description
04h	Number of times the HTC trip point is crossed
08h	Number of clocks when STC trip point active
10h	Number of times the STC trip point is crossed



20h	Number of clocks HTC P-state is inactive.
40h	Number of clocks HTC P-state is active

### EventSelect 0E9h 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
01h	IO to IO
02h	IO to Mem
04h	CPU to IO
08h	CPU to Mem
10h	To remote node
20h	To local node
40h	From remote node
80h	From local node

### EventSelect 0EAh 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
01h	Victim Block (Writeback)
04h	Read Block (Dcache load miss refill)
08h	Read Block Shared (Icache refill)
10h	Read Block Modified (Dcache store miss refill)
20h	Change-to-Dirty (first store to clean block already in cache)

### EventSelect 0EBh 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 [EventSelect 0ECh](#), which provides a separate measure of Hostbridge accesses.

UnitMask	Description	Typical Usage
01h	Non-Posted SzWr Byte (1-32 bytes)	Legacy or mapped IO, typically 1-4 bytes
02h	Non-Posted SzWr DW (1-16 dwords)	Legacy or mapped IO, typically 1 DWORD
04h	Posted SzWr Byte (1-32 bytes)	Subcache-line DMA writes, size varies; also flushes of partially-filled Write Combining buffer
08h	Posted SzWr DW (1-16 dwords)	Block-oriented DMA writes, often cache-line sized; also processor Write Combining buffer flushes
10h	SzRd Byte (4 bytes)	Legacy or mapped IO
20h	SzRd DW (1-16 dwords)	Block-oriented DMA reads, typically cache-line size

### EventSelect 0ECh 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 [EventSelect 06Ch](#), System Read Responses.)

**Upstream requests:** The upstream read and write events reflect requests originating from a device on a local IO link. The two read events allow display refresh traffic in a UMA system to be measured separately from other DMA activity. Display refresh traffic is typically dominated by 64-byte transfers. Non-display-related DMA accesses may be anywhere from 1 to 64 bytes in size, but may be dominated by a particular size such as 32 or 64 bytes, depending on the nature of the devices.

UnitMask	Description
01h	Probe miss
02h	Probe hit clean
04h	Probe hit dirty without memory cancel (probed by Sized Write or Change2Dirty)
08h	Probe hit dirty with memory cancel (probed by DMA read or cache refill request)
10h	Upstream display refresh/ISOC reads
20h	Upstream non-display refresh reads
40h	Upstream ISOC writes
80h	Upstream non-ISOC writes

### EventSelect 0EEh GART Events

These events reflect GART activity, and in particular allow one to calculate the GART TLB miss ratio as GART\_miss\_count divided by GART\_aperture\_hit\_count. GART aperture accesses are typically from IO devices as opposed to the processor, and generally from a 3D graphics accelerator, but can be from other devices when the GART is used as an IOMMU.

UnitMask	Description
01h	GART aperture hit on access from CPU
02h	GART aperture hit on access from IO
04h	GART miss
08h	GART/DEV Request hit table walk in progress
10h	DEV hit
20h	DEV miss
40h	DEV error
80h	GART/DEV multiple table walk in progress

### EventSelect 1F0h 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. [EventSelect 065h](#) 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. [EventSelect 065h](#) may also be useful in factoring out WC flushes when comparing these events with the Upstream Requests component of [EventSelect 06Ch](#).

UnitMask	Description
01h	Write requests sent to the DCT
02h	Read requests (including prefetch requests) sent to the DCT
04h	Prefetch requests sent to the DCT

08h	32 Bytes Sized Writes
10h	64 Bytes Sized Writes
20h	32 Bytes Sized Reads
40h	64 Byte Sized Reads
80h	Read requests sent to the DCT while writes requests are pending in the DCT

### 3.14.8 Crossbar Events

#### **EventSelect 1E0h 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
01h	From Local node to Node 0
02h	From Local node to Node 1
04h	From Local node to Node 2
08h	From Local node to Node 3
10h	From Local node to Node 4
20h	From Local node to Node 5
40h	From Local node to Node 6
80h	From Local node to Node 7

#### **EventSelect 1E1h 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
01h	From Local node to Node 0
02h	From Local node to Node 1
04h	From Local node to Node 2
08h	From Local node to Node 3
10h	From Local node to Node 4
20h	From Local node to Node 5
40h	From Local node to Node 6
80h	From Local node to Node 7

#### **EventSelect 1E2h 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 [EventSelect 1E3h](#) to determine the average latency for the command type.

UnitMask	Description
01h	Read block
02h	Read block shared
04h	Read block modified
08h	Change-to-Dirty
10h	From Local node to Node 0
20h	From Local node to Node 1
40h	From Local node to Node 2
80h	From Local node to Node 3

### **EventSelect 1E3h CPU Read Command Requests to Target Node 0-3**

This event counts the number of requests that a latency measurement is made for using [EventSelect 1E2h](#). 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
01h	Read block
02h	Read block shared
04h	Read block modified
08h	Change-to-Dirty
10h	From Local node to Node 0
20h	From Local node to Node 1
40h	From Local node to Node 2
80h	From Local node to Node 3

### **EventSelect 1E4h 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 [EventSelect 1E5h](#) to determine the average latency for the command type.

UnitMask	Description
01h	Read block
02h	Read block shared
04h	Read block modified
08h	Change-to-Dirty
10h	From Local node to Node 4
20h	From Local node to Node 5

40h	From Local node to Node 6
80h	From Local node to Node 7

#### **EventSelect 1E5h CPU Read Command Requests to Target Node 4-7**

This event counts the number of requests that a latency measurement is made for using [EventSelect 1E4h](#). 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
01h	Read block
02h	Read block shared
04h	Read block modified
08h	Change-to-Dirty
10h	From Local node to Node 4
20h	From Local node to Node 5
40h	From Local node to Node 6
80h	From Local node to Node 7

#### **EventSelect 1E6h 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 [EventSelect 1E7h](#) to determine the average latency for the command type.

UnitMask	Description
01h	Read Sized
02h	Write Sized
04h	Victim Block
08h	Node Group Select. 0=Nodes 0-3. 1= Nodes 4-7.
10h	From Local node to Node 0/4
20h	From Local node to Node 1/5
40h	From Local node to Node 2/6
80h	From Local node to Node 3/7

#### **EventSelect 1E7h CPU Requests to Target Node 0-3/4-7**

This event counts the number of requests that a latency measurement is made for using [EventSelect 1E6h](#). 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
01h	Read Sized
02h	Write Sized
04h	Victim Block
08h	Node Group Select. 0=Nodes 0-3. 1= Nodes 4-7.
10h	From Local node to Node 0/4
20h	From Local node to Node 1/5
40h	From Local node to Node 2/6
80h	From Local node to Node 3/7

### 3.14.9 Link Events

#### **EventSelect 0F6h HyperTransport™ Link 0 Transmit Bandwidth**

#### **EventSelect 0F7h HyperTransport™ Link 1 Transmit Bandwidth**

#### **EventSelect 0F8h HyperTransport™ Link 2 Transmit Bandwidth**

#### **EventSelect 1F9h HyperTransport™ Link 3 Transmit Bandwidth**

The number of DWORDs transmitted (or unused, in the case of NOPs) on the outgoing side of the HyperTransport™ links. The sum of UnitMask[5:0] directly reflects the maximum transmission rate of the link. Link utilization may be calculated by dividing the combined Command, Address extension, Data, Buffer Release and Per packet CRC count (UnitMask 02Fh) by that value plus the Nop count (UnitMask 10h). Bandwidth in terms of bytes per unit time for any one component or combination of components is calculated by multiplying the count by four and dividing by elapsed time.

The Data event provides a direct indication of the flow of data around the system. Translating this link-based view into a source/target node based view requires knowledge of the system layout (i.e. which links connect to which nodes).

UnitMask[7] specifies the sublink to count if the link is ungangged.

UnitMask	Description
01h	Command DWORD sent
02h	Data DWORD sent
04h	Buffer release DWORD sent
08h	Nop DW sent (idle)
10h	Address DWORD sent
20h	Per packet CRC sent
80h	SubLink Mask

### 3.14.10 L3 Cache Events

#### **EventSelect 4E0h Read Request to L3 Cache**

This event tracks the read requests from each core to the L3 cache including read requests that are cancelled. The core tracked is selected using UnitMask[7:4]. One or more cores must be selected. To determine the total number of read requests from one core, select only a single core using UnitMask[7:4] and set UnitMask[2:0] to 111b.



Revision C and earlier:

UnitMask	Description
01h	Read Block Exclusive (Data cache read)
02h	Read Block Shared (Instruction cache read)
04h	Read Block Modify
10h	Core 0 Select
20h	Core 1 Select
40h	Core 2 Select
80h	Core 3 Select

Revision D and later:

UnitMask	Description
01h	Read Block Exclusive (Data cache read)
02h	Read Block Shared (Instruction cache read)
04h	Read Block Modify
00h	Core 0 Select
10h	Core 1 Select
20h	Core 2 Select
30h	Core 3 Select
40h	Core 4 Select
50h	Core 5 Select
F0h	Any core

### **EventSelect 4E1h L3 Cache Misses**

This event counts the number of L3 cache misses for accesses from each core. The core tracked is selected using UnitMask[7:4]. One or more cores must be selected. To determine the total number of cache misses from one core, select only a single core using UnitMask[7:4] and set UnitMask[2:0] to 111b. The approximate number of L3 hits can be determined by subtracting this event from [EventSelect 4E0h](#).

Revision C and earlier:

UnitMask	Description
01h	Read Block Exclusive (Data cache read)
02h	Read Block Shared (Instruction cache read)
04h	Read Block Modify
10h	Core 0 Select
20h	Core 1 Select
40h	Core 2 Select
80h	Core 3 Select

Revision D and later:

UnitMask	Description
01h	Read Block Exclusive (Data cache read)
02h	Read Block Shared (Instruction cache read)
04h	Read Block Modify
00h	Core 0 Select
10h	Core 1 Select
20h	Core 2 Select
30h	Core 3 Select
40h	Core 4 Select
50h	Core 5 Select
F0h	Any core

#### **EventSelect 4E2h L3 Fills caused by L2 Evictions**

This event counts the number of L3 fills caused by L2 evictions. The core tracked is selected using UnitMask[7:4]. One or more cores must be selected.

Revision C and earlier:

UnitMask	Description
01h	Shared
02h	Exclusive
04h	Owned
08h	Modified
10h	Core 0 Select
20h	Core 1 Select
40h	Core 2 Select
80h	Core 3 Select

Revision D and later:

UnitMask	Description
01h	Shared
02h	Exclusive
04h	Owned
08h	Modified
00h	Core 0 Select
10h	Core 1 Select
20h	Core 2 Select
30h	Core 3 Select
40h	Core 4 Select

50h	Core 5 Select
F0h	Any core

### EventSelect 4E3h L3 Evictions

This event counts the state of the L3 lines when they are evicted from the L3 cache.

UnitMask	Description
01h	Shared
02h	Exclusive
04h	Owned
08h	Modified

### EventSelect 4EDh Non-cancelled L3 Read Requests

Revision C and earlier: Reserved. Revision D and later: This event tracks all read requests from each core to the L3 cache that are not cancelled. The core tracked is selected using UnitMask[7:4]. One or more cores must be selected. To determine the total number of read requests from one core, select only a single core using UnitMask[7:4] and set UnitMask[2:0] to 111b.

UnitMask	Description
01h	RdBlk
02h	RdBlkS
04h	RdBlkM
00h	Core 0 Select
10h	Core 1 Select
20h	Core 2 Select
30h	Core 3 Select
40h	Core 4 Select
50h	Core 5 Select
F0h	Any core

## 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.

191	SMMFEC0: SMM IO Trap Offset	234	F2x[1, 0][5C:40]: DRAM CS Base Address Registers
192	SMMFEC4: Local SMI Status	236	F2x[1, 0][6C:60]: DRAM CS Mask Registers
193	SMMFEC8: SMM IO Restart Byte	236	F2x[1, 0]78: DRAM Control Register
193	SMMFEC9: Auto Halt Restart Offset	237	F2x[1, 0]7C: DRAM Initialization Register
193	SMMFECA: NMI Mask	238	F2x[1, 0]80: DRAM Bank Address Mapping Register
193	SMMFED8: SMM SVM State	240	F2x[1, 0]84: DRAM MRS Register
194	SMMFEFC: SMM-Revision Identifier	242	F2x[1, 0]88: DRAM Timing Low Register
194	SMMFF00: SMM Base Address Register (SMM_BASE)	245	F2x[1, 0]8C: DRAM Timing High Register
200	IOCF8: IO-Space Configuration Address Register	248	F2x[1, 0]90: DRAM Configuration Low Register
201	IOCF8: IO-Space Configuration Address Register	250	F2x[1, 0]94: DRAM Configuration High Register
201	F0x00: Device/Vendor ID Register	254	F2x[1, 0]98: DRAM Controller Additional Data Offset Register
201	F0x04: Status/Command Register	255	F2x[1, 0]9C: DRAM Controller Additional Data Port
202	F0x08: Class Code/Revision ID Register	273	F2x[1, 0]A0: DRAM Controller Miscellaneous Register
202	F0x0C: Header Type Register	273	F2x[1, 0]A4: DRAM Controller Temperature Throttle Register
202	F0x34: Capabilities Pointer Register	274	F2x[1, 0]A8: DRAM Controller Miscellaneous Register 2
202	F0x[5C:40]: Routing Table Registers	275	F2xAC: DRAM Controller Temperature Status Register
203	F0x60: Node ID Register	275	F2x[1, 0]F0: DRAM Controller Extra Data Offset Register
203	F0x64: Unit ID Register	276	F2x[1, 0]F4: DRAM Controller Extra Data Port
204	F0x68: Link Transaction Control Register	277	F2xF8: P-state Power Information Register
206	F0x6C: Link Initialization Control Register	277	F2xFC: P-state Power Information Register 2
207	F0x[E0, C0, A0, 80]: Link Capabilities Registers	277	F2x10C: Swap Interleaved Region Base/Limit Register
208	F0x[E4, C4, A4, 84]: Link Control Registers	278	F2x110: DRAM Controller Select Low Register
210	F0x[E8, C8, A8, 88]: Link Frequency/Revision Registers	280	F2x114: DRAM Controller Select High Register
211	F0x[EC, CC, AC, 8C]: Link Feature Capability Registers	280	F2x118: Memory Controller Configuration Low Register
212	F0x[F0, D0, B0, 90]: Link Base Channel Buffer Count Registers	281	F2x11C: Memory Controller Configuration High Register
214	F0x[F4, D4, B4, 94]: Link Isochronous Channel Buffer Count Registers	283	F2x1B0: Extended Memory Controller Configuration Low Register
214	F0x[F8, D8, B8, 98]: Link Type Registers	285	F3x00: Device/Vendor ID Register
215	F0x[FC, DC, BC, 9C]: Link Frequency Extension Registers	285	F3x04: Status/Command Register
215	F0x[11C, 118, 114, 110]: Link Clumping Enable Registers	286	F3x08: Class Code/Revision ID Register
216	F0x[12C, 128, 124, 120]: Sublink 1 Clumping Enable Registers	286	F3x0C: Header Type Register
216	F0x[14C:130]: Link Retry Registers	286	F3x34: Capability Pointer Register
217	F0x150: Link Global Retry Control Register	286	F3x40: MCA NB Control Register
218	F0x160: Extended Node ID Register	289	F3x44: MCA NB Configuration Register
218	F0x164: Coherent Link Traffic Distribution Register	292	F3x48: MCA NB Status Low Register
219	F0x168: Extended Link Transaction Control Register	298	F3x4C: MCA NB Status High Register
219	F0x16C: Link Global Extended Control Register	300	F3x50: MCA NB Address Low Register
221	F0x[18C:170]: Link Extended Control Registers	304	F3x54: MCA NB Address High Register
223	F0x1A0: Link Initialization Status Register	304	F3x58: Scrub Rate Control Register
223	F0x1E0: Coherent Link Pair Traffic Distribution Register	305	F3x5C: DRAM Scrub Address Low Register
224	F1x00: Device/Vendor ID Register	306	F3x64: Hardware Thermal Control (HTC) Register
224	F1x08: Class Code/Revision ID Register	306	F3x60: DRAM Scrub Address High Register
224	F1x0C: Header Type Register	307	F3x68: Software Thermal Control (STC) Register
224	F1x[1, 0][7C:40]: DRAM Base/Limit Registers	308	F3x6C: Data Buffer Count Register
226	F1x[BC:80]: Memory Mapped IO Base/Limit Registers	309	F3x70: SRI to XBAR Command Buffer Count Register
227	F1x[DC:C0]: IO-Space Base/Limit Registers	309	F3x74: XBAR to SRI Command Buffer Count Register
228	F1x[EC:E0]: Configuration Map Registers	310	F3x78: MCT to XBAR Buffer Count Register
229	F1xF0: DRAM Hole Address Register	311	F3x7C: Free List Buffer Count Register
230	F1xF4: VGA Enable Register	312	F3x[84:80]: ACPI Power State Control Registers
230	F1x110: Extended Address Map Control Register	314	F3x[8C:88]: NB Configuration High, Low Registers
231	F1x114: Extended Address Map Data Port	314	F3x90: GART Aperture Control Register
232	F1x120: DRAM Base System Address Register	315	F3x94: GART Aperture Base Register
233	F1x124: DRAM Limit System Address Register	315	F3x98: GART Table Base Register
233	F2x00: Device/Vendor ID Register	315	F3x9C: GART Cache Control Register
233	F2x08: Class Code/Revision ID Register	316	F3xA0: Power Control Miscellaneous Register
233	F2x0C: Header Type Register	317	F3xA4: Reported Temperature Control Register

318	F3xA8: Popup P-state Register	373	APICA0: Processor Priority Register
319	F3xB0: On-Line Spare Control Register	374	APICB0: End of Interrupt Register
320	F3xC4: SBI P-state Limit Register	374	APICCO: Remote Read Register
320	F3xD4: Clock Power/Timing Control 0 Register	374	APICD0: Logical Destination Register
323	F3xD8: Clock Power/Timing Control 1 Register	374	APICE0: Destination Format Register
325	F3xDC: Clock Power/Timing Control 2 Register	374	APICF0: Spurious Interrupt Vector Register
326	F3xE4: Thermtrip Status Register	375	APIC[170:100]: In-Service Registers
327	F3xE8: Northbridge Capabilities Register	375	APIC[1F0:180]: Trigger Mode Registers
328	F3xF0: DEV Capability Header Register	375	APIC[270:200]: Interrupt Request Registers
329	F3xF4: DEV Function/Index Register	376	APIC280: Error Status Register
329	F3xF8: DEV Data Port	376	APIC300: Interrupt Command Register Low
332	F3xFC: CUID Family/Model Register	378	APIC310: Interrupt Command Register High
332	F3x10C: Boost Offset	378	APIC320: Timer Local Vector Table Entry
333	F3x138: DCT0 Bad Symbol Identification Register	378	APIC330: Thermal Local Vector Table Entry
333	F3x13C: DCT1 Bad Symbol Identification Register	379	APIC340: Performance Counter Vector Table Entry
333	F3x140: SRI to XCS Token Count Register	379	APIC350: Local Interrupt 0 (Legacy INTR) Local Vector Table Entry
334	F3x144: MCT to XCS Token Count Register	380	APIC360: Local Interrupt 1 (Legacy NMI) Local Vector Table Entry
335	F3x1[54, 50, 4C, 48]: Link to XCS Token Count Registers	380	APIC370: Error Local Vector Table Entry
335	F3x158: Link to XCS Token Count Registers	380	APIC380: Timer Initial Count Register
336	F3x1[78, 70, 68, 60]: NB Machine Check Misc (Thresholding) Registers	380	APIC390: Timer Current Count Register
337	F3x180: Extended NB MCA Configuration Register	380	APIC3E0: Timer Divide Configuration Register
338	F3x188: NB Extended Configuration Low Register	381	APIC400: Extended APIC Feature Register
338	F3x190: Downcore Control Register	381	APIC410: Extended APIC Control Register
338	F3x1A0: L3 Buffer Count Register	382	APIC420: Specific End Of Interrupt Register
339	F3x1C4: L3 Cache Parameter Register	382	APIC[4F0:480]: Interrupt Enable Registers
340	F3x1CC: IBS Control Register	382	APIC[530:500]: Extended Interrupt [3:0] Local Vector Table Registers
340	F3x1D4: Probe Filter Control Register	383	CUID Fn[8000_0000, 0000_0000]: AMD Authentic Identifier
341	F3x1E4: SBI Control Register	383	CUID Fn[8000_0001, 0000_0001]_EAX: Family, Model, Stepping Identifiers
342	F3x1E8: SBI Address Register	384	CUID Fn0000_0001_EBX: LocalApicId, LogicalProcessorCount, CLFlush, 8BitBrandId
342	F3x1EC: SBI Data Register	384	CUID Fn8000_0001_EBX: BrandId Identifier
344	F3x1F0: Product Information Register	384	CUID Fn0000_0001_ECX: Feature Identifiers
345	F3x1FC: Product Information Register	385	CUID Fn8000_0001_ECX: Feature Identifiers
346	F4x00: Device/Vendor ID Register	385	CUID Fn[8000_0001, 0000_0001]_EDX: Feature Identifiers
346	F4x04: Status/Command Register	387	CUID Fn0000_000[4, 3, 2]: Reserved
346	F4x08: Class Code/Revision ID Register	387	CUID Fn8000_000[4, 3, 2]: Processor Name String Identifier
346	F4x0C: Header Type Register	387	CUID Fn0000_0005: Monitor/MWait
346	F4x34: Capabilities Pointer Register	387	CUID Fn8000_0005: TLB and L1 Cache Identifiers
347	F4x[E0, C0, A0, 80]: Sublink 1 Capability Registers	388	CUID Fn0000_0006: Power Management
347	F4x[E4, C4, A4, 84]: Sublink 1 Control Registers	388	CUID Fn8000_0006: L2/L3 Cache and L2 TLB Identifiers
347	F4x[E8, C8, A8, 88]: Sublink 1 Frequency/Revision Registers	389	CUID Fn8000_0007: Advanced Power Management Information
347	F4x[EC, CC, AC, 8C]: Sublink 1 Feature Capability Registers	390	CUID Fn8000_0008: Address Size And Physical Core Count Information
347	F4x[F0, D0, B0, 90]: Sublink 1 Base Channel Buffer Count Registers	390	CUID Fn8000_0009: Reserved
347	F4x[F4, D4, B4, 94]: Sublink 1 Isochronous Channel Buffer Count Registers	390	CUID Fn8000_000A: SVM Revision and Feature Identification
347	F4x[F8, D8, B8, 98]: Sublink 1 Link Type Registers	391	CUID Fn8000_00[18:0B]: Reserved
347	F4x[FC, DC, BC, 9C]: Sublink 1 Link Frequency Extension Registers	391	CUID Fn8000_0019: TLB 1GB Page Identifiers
347	F4x15C: Core Performance Boost Control	391	CUID Fn8000_001A: Performance Optimization Identifiers
348	F4x16C: Cstate Boost Control	391	CUID Fn8000_001B: Instruction Based Sampling Identifiers
348	F4x1[98, 90, 88, 80]: Link Phy Offset Registers	392	MSR0000_0000: Load-Store MCA Address Register
349	F4x1[9C, 94, 8C, 84]: Link Phy Data Port	392	MSR0000_0001: Load-Store MCA Status Register
371	F4x1C4: L3 Power Control Register	392	MSR0000_0010: Time Stamp Counter Register (TSC)
371	F4x1C8: L3 Hit Statistics Register	392	MSR0000_001B: APIC Base Address Register (APIC_BAR)
372	F4x1[F0:E0]: P-state Specification Registers	393	MSR0000_002A: Cluster ID Register (EBL_CR_POWERON)
372	F4x1F4: Northbridge P-state Specification Register	393	MSR0000_00E7: Max Performance Frequency Clock Count (MPERF)
373	APIC20: APIC ID Register	393	MSR0000_00E8: Actual Performance Frequency Clock Count (APERF)
373	APIC30: APIC Version Register	393	MSR0000_00FE: MTRR Capabilities Register (MTRRcap)
373	APIC80: Task Priority Register		
373	APIC90: Arbitration Priority Register		

393	MSR0000_0174: SYSENTER CS Register (SYSENTER_CS)	412	MSRC000_0101: GS Base Register (GS_BASE)
394	MSR0000_0175: SYSENTER ESP Register (SYSENTER_ESP)	412	MSRC000_0102: Kernel GS Base Register (KernelGSbase)
394	MSR0000_0176: SYSENTER EIP Register (SYSENTER_EIP)	412	MSRC000_0103: Auxiliary Time Stamp Counter Register (TSC_AUX)
394	MSR0000_0179: Global Machine Check Capabilities Register (MCG_CAP)	412	MSRC000_04[0A:08]: Machine Check Misc 4 (Thresholding) Registers 1 to 3 (MC4_MISC[3:1])
394	MSR0000_017A: Global Machine Check Status Register (MCG_STAT)	413	MSRC000_04[0F:0B]: Reserved
395	MSR0000_017B: Global Machine Check Exception Reporting Control Register (MCG_CTL)	413	MSRC001_00[03:00]: Performance Event Select Register (PERF_CTL[3:0])
395	MSR0000_01D9: Debug Control Register (DBG_CTL_MSR)	414	MSRC001_00[07:04]: Performance Event Counter Registers (PERF_CTR[3:0])
395	MSR0000_01DB: Last Branch From IP Register (BR_FROM)	415	MSRC001_0010: System Configuration Register (SYS_CFG)
395	MSR0000_01DC: Last Branch To IP Register (BR_TO)	416	MSRC001_0015: Hardware Configuration Register (HWCR)
396	MSR0000_01DD: Last Exception From IP Register	418	MSRC001_00[18, 16]: IO Range Registers Base (IORR_BASE[1:0])
396	MSR0000_01DE: Last Exception To IP Register	418	MSRC001_00[19, 17]: IO Range Registers Mask (IORR_MASK[1:0])
396	MSR0000_02[0F:00]: Variable-Size MTRRs (MTRRphysBasen and MTRRphysMaskn)	418	MSRC001_001A: Top Of Memory Register (TOP_MEM)
397	MSR0000_02[6F:68, 59, 58, 50]: Fixed-Size MTRRs (MTRRfixn)	418	MSRC001_001D: Top Of Memory 2 Register (TOM2)
398	MSR0000_0277: Page Attribute Table Register (PAT)	419	MSRC001_001F: Northbridge Configuration Register (NB_CFG)
399	MSR0000_02FF: MTRR Default Memory Type Register (MTRRdefType)	420	MSRC001_0022: Machine Check Exception Redirection Register
399	MSR0000_0400: DC Machine Check Control Register (MC0_CTL)	420	MSRC001_00[35:30]: Processor Name String Registers
400	MSR0000_0401: DC Machine Check Status Register (MC0_STATUS)	421	MSRC001_003E: Hardware Thermal Control Register (HTC)
402	MSR0000_0402: DC Machine Check Address Register (MC0_ADDR)	421	MSRC001_00[49:44]: Machine Check Control Mask Registers (MCi_CTL_MASK)
403	MSR0000_0403: DC Machine Check Miscellaneous Register (MC0_MISC)	421	MSRC001_00[53:50]: IO Trap Registers (SMI_ON_IO_TRAP_[3:0])
403	MSR0000_0404: IC Machine Check Control Register (MC1_CTL)	422	MSRC001_0054: IO Trap Control Register (SMI_ON_IO_TRAP_CTL_STS)
404	MSR0000_0405: IC Machine Check Status Register (MC1_STATUS)	422	MSRC001_0055: Interrupt Pending and CMP-Halt Register
405	MSR0000_0406: IC Machine Check Address Register (MC1_ADDR)	423	MSRC001_0056: SMI Trigger IO Cycle Register
405	MSR0000_0407: IC Machine Check Miscellaneous Register (MC1_MISC)	424	MSRC001_0058: MMIO Configuration Base Address Register
405	MSR0000_0408: BU Machine Check Control Register (MC2_CTL)	424	MSRC001_0061: P-State Current Limit Register
406	MSR0000_0409: BU Machine Check Status Register (MC2_STATUS)	425	MSRC001_0062: P-State Control Register
407	MSR0000_040A: BU Machine Check Address Register (MC2_ADDR)	425	MSRC001_0063: P-State Status Register
408	MSR0000_040B: BU Machine Check Miscellaneous Register (MC2_MISC)	425	MSRC001_00[68:64]: P-State [4:0] Registers
408	MSR0000_040C: LS Machine Check Control Register (MC3_CTL)	426	MSRC001_0070: COFVID Control Register
408	MSR0000_040D: LS Machine Check Status Register (MC3_STATUS)	427	MSRC001_0071: COFVID Status Register
408	MSR0000_040E: LS Machine Check Address Register (MC3_ADDR)	428	MSRC001_0072: SBI P-state Limit Register
408	MSR0000_040F: LS Machine Check Miscellaneous Register (MC3_MISC)	428	MSRC001_0073: C-state Base Address
408	MSR0000_0410: NB Machine Check Control Register (MC4_CTL)	428	MSRC001_0074: CPU Watchdog Timer Register (CpuWdTmrCfg)
409	MSR0000_0411: NB Machine Check Status Register (MC4_STATUS)	429	MSRC001_0111: SMM Base Address Register (SMM_BASE)
409	MSR0000_0412: NB Machine Check Address Register (MC4_ADDR)	429	MSRC001_0112: SMM TSeg Base Address Register (SMMAddr)
409	MSR0000_0413: NB Machine Check Misc (Thresholding) Register (MC4_MISC0)	430	MSRC001_0113: SMM TSeg Mask Register (SMMMask)
409	MSR0000_0414: FR Machine Check Control Register (MC5_CTL)	431	MSRC001_0114: Virtual Machine Control Register (VM_CR)
410	MSR0000_0415: FR Machine Check Status Register (MC5_STATUS)	431	MSRC001_0115: IGNNE Register (IGNNE)
410	MSR0000_0416: FR Machine Check Address Register (MC5_ADDR)	431	MSRC001_0116: SMM Control Register (SMM_CTL)
410	MSR0000_0417: FR Machine Check Miscellaneous Register (MC5_MISC)	432	MSRC001_0117: Virtual Machine Host Save Physical Address Register (VM_HSAVE_PA)
410	MSRC000_0080: Extended Feature Enable Register (EFER)	432	MSRC001_0118: SVM Lock Key
411	MSRC000_0081: SYSCALL Target Address Register (STAR)	432	MSRC001_011A: Local SMI Status
411	MSRC000_0082: Long Mode SYSCALL Target Address Register (STAR64)	433	MSRC001_0140: OS Visible Work-around MSR0 (OSVW_ID_Length)
411	MSRC000_0083: Compatibility Mode SYSCALL Target Address Register (STARCOMPAT)	433	MSRC001_0141: OS Visible Work-around MSR1 (OSVW Status)
411	MSRC000_0084: SYSCALL Flag Mask Register (SYSCALL_FLAG_MASK)	433	MSRC001_1004: CPUID Features Register (Features)
412	MSRC000_0100: FS Base Register (FS_BASE)	433	MSRC001_1005: Extended CPUID Features Register (ExtFeatures)
		433	MSRC001_100C: Node ID
		434	MSRC001_1021: Instruction Cache Configuration Register (IC_CFG)
		434	MSRC001_1022: Data Cache Configuration Register (DC_CFG)
		434	MSRC001_1023: Bus Unit Configuration Register (BU_CFG)
		435	MSRC001_102A: Bus Unit Configuration 2 Register (BU_CFG2)
		436	MSRC001_1030: IBS Fetch Control Register (IbsFetchCtl)
		437	MSRC001_1031: IBS Fetch Linear Address Register (IbsFetchLinAd)

437	MSRC001_1032: IBS Fetch Physical Address Register (IbsFetchPhysAd)	451	EventSelect 085h: L1 ITLB Miss, L2 ITLB Miss
437	MSRC001_1033: IBS Execution Control Register (IbsOpCtl)	451	EventSelect 086h: Pipeline Restart Due to Instruction Stream Probe
438	MSRC001_1034: IBS Op Logical Address Register (IbsOpRip)	451	EventSelect 087h: Instruction Fetch Stall
438	MSRC001_1035: IBS Op Data Register (IbsOpData)	451	EventSelect 088h: Return Stack Hits
439	MSRC001_1036: IBS Op Data 2 Register (IbsOpData2)	451	EventSelect 089h: Return Stack Overflows
439	MSRC001_1037: IBS Op Data 3 Register (IbsOpData3)	451	EventSelect 08Bh: Instruction Cache Victims
440	MSRC001_1038: IBS DC Linear Address Register (IbsDcLinAd)	452	EventSelect 08Ch: Instruction Cache Lines Invalidated
440	MSRC001_1039: IBS DC Physical Address Register (IbsDcPhysAd)	452	EventSelect 099h: ITLB Reloads
440	MSRC001_103A: IBS Control Register	452	EventSelect 09Ah: ITLB Reloads Aborted
441	EventSelect 000h: Dispatched FPU Operations	452	EventSelect 0C0h: Retired Instructions
441	EventSelect 001h: Cycles in which the FPU is Empty	452	EventSelect 0C1h: Retired uops
441	EventSelect 002h: Dispatched Fast Flag FPU Operations	452	EventSelect 0C2h: Retired Branch Instructions
442	EventSelect 003h: Retired SSE Operations	452	EventSelect 0C3h: Retired Mispredicted Branch Instructions
442	EventSelect 004h: Retired Move Ops	452	EventSelect 0C4h: Retired Taken Branch Instructions
442	EventSelect 005h: Retired Serializing Ops	452	EventSelect 0C5h: Retired Taken Branch Instructions Mispredicted
442	EventSelect 006h: Number of Cycles that a Serializing uop is in the FP Scheduler	452	EventSelect 0C6h: Retired Far Control Transfers
443	EventSelect 020h: Segment Register Loads	453	EventSelect 0C7h: Retired Branch Resyncs
443	EventSelect 021h: Pipeline Restart Due to Self-Modifying Code	453	EventSelect 0C8h: Retired Near Returns
443	EventSelect 022h: Pipeline Restart Due to Probe Hit	453	EventSelect 0C9h: Retired Near Returns Mispredicted
443	EventSelect 023h: LS Buffer 2 Full	453	EventSelect 0CAh: Retired Indirect Branches Mispredicted
443	EventSelect 024h: Locked Operations	453	EventSelect 0CBh: Retired MMX™/FP Instructions
444	EventSelect 026h: Retired CLFLUSH Instructions	453	EventSelect 0CCh: Retired Fastpath Double Op Instructions
444	EventSelect 027h: Retired CPUID Instructions	453	EventSelect 0CDh: Interrupts-Masked Cycles
444	EventSelect 02Ah: Cancelled Store to Load Forward Operations	453	EventSelect 0CEh: Interrupts-Masked Cycles with Interrupt Pending
444	EventSelect 02Bh: SMIs Received	454	EventSelect 0CFh: Interrupts Taken
444	EventSelect 040h: Data Cache Accesses	454	EventSelect 0D0h: Decoder Empty
444	EventSelect 041h: Data Cache Misses	454	EventSelect 0D1h: Dispatch Stalls
444	EventSelect 042h: Data Cache Refills from L2 or Northbridge	454	EventSelect 0D2h: Dispatch Stall for Branch Abort to Retire
445	EventSelect 043h: Data Cache Refills from the Northbridge	454	EventSelect 0D3h: Dispatch Stall for Serialization
445	EventSelect 044h: Data Cache Lines Evicted	454	EventSelect 0D4h: Dispatch Stall for Segment Load
446	EventSelect 045h: L1 DTLB Miss and L2 DTLB Hit	454	EventSelect 0D5h: Dispatch Stall for Reorder Buffer Full
446	EventSelect 046h: L1 DTLB and L2 DTLB Miss	454	EventSelect 0D6h: Dispatch Stall for Reservation Station Full
446	EventSelect 047h: Misaligned Accesses	454	EventSelect 0D7h: Dispatch Stall for FPU Full
446	EventSelect 048h: Microarchitectural Late Cancel of an Access	454	EventSelect 0D8h: Dispatch Stall for LS Full
446	EventSelect 049h: Microarchitectural Early Cancel of an Access	455	EventSelect 0D9h: Dispatch Stall Waiting for All Quiet
446	EventSelect 04Ah: Single-bit ECC Errors Recorded by Scrubber	455	EventSelect 0DAh: Dispatch Stall for Far Transfer or Resync to Retire
446	EventSelect 04Bh: Prefetch Instructions Dispatched	455	EventSelect 0DBh: FPU Exceptions
447	EventSelect 04Ch: DCACHE Misses by Locked Instructions	455	EventSelect 0DCh: DR0 Breakpoint Matches
447	EventSelect 04Dh: L1 DTLB Hit	455	EventSelect 0DDh: DR1 Breakpoint Matches
447	EventSelect 052h: Ineffective Software Prefetches	455	EventSelect 0DEh: DR2 Breakpoint Matches
447	EventSelect 054h: Global TLB Flushes	455	EventSelect 0DFh: DR3 Breakpoint Matches
447	EventSelect 065h: Memory Requests by Type	455	EventSelect 1C0h: Retired x87 Floating Point Operations
448	EventSelect 067h: Data Prefetcher	455	EventSelect 1CFh: IBS Ops Tagged
448	EventSelect 068h: MAB Requests	456	EventSelect 1D3h: LFENCE Instructions Retired
448	EventSelect 069h: MAB Wait Cycles	456	EventSelect 1D4h: SFENCE Instructions Retired
448	EventSelect 06Ch: Northbridge Read Responses by Coherency State	456	EventSelect 1D5h: MFENCE Instructions Retired
449	EventSelect 06Dh: Octwords Written to System	456	EventSelect 0E0h: DRAM Accesses
449	EventSelect 076h: CPU Clocks not Halted	456	EventSelect 0E1h: DRAM Controller Page Table Overflows
449	EventSelect 07Dh: Requests to L2 Cache	457	EventSelect 0E2h: Memory Controller DRAM Command Slots Missed
449	EventSelect 07Eh: L2 Cache Misses	457	EventSelect 0E3h: Memory Controller Turnarounds
450	EventSelect 07Fh: L2 Fill/Writeback	457	EventSelect 0E4h: Memory Controller Bypass Counter Saturation
450	EventSelect 165h: Page Size Mismatches	457	EventSelect 0E8h: Thermal Status
450	EventSelect 080h: Instruction Cache Fetches	458	EventSelect 0E9h: CPU/IO Requests to Memory/IO
450	EventSelect 081h: Instruction Cache Misses	458	EventSelect 0EAh: Cache Block Commands
451	EventSelect 082h: Instruction Cache Refills from L2	459	EventSelect 0EBh: Sized Commands
451	EventSelect 083h: Instruction Cache Refills from System	459	EventSelect 0ECh: Probe Responses and Upstream Requests
451	EventSelect 084h: L1 ITLB Miss, L2 ITLB Hit	460	EventSelect 0EEh: GART Events
		460	EventSelect 1F0h: Memory Controller Requests
		461	EventSelect 1E0h: CPU to DRAM Requests to Target Node



461 EventSelect 1E1h: IO to DRAM Requests to Target Node  
461 EventSelect 1E2h: CPU Read Command Latency to Target Node 0-3  
462 EventSelect 1E3h: CPU Read Command Requests to Target Node 0-3  
462 EventSelect 1E4h: CPU Read Command Latency to Target Node 4-7  
463 EventSelect 1E5h: CPU Read Command Requests to Target Node 4-7  
463 EventSelect 1E6h: CPU Command Latency to Target Node 0-3/4-7  
463 EventSelect 1E7h: CPU Requests to Target Node 0-3/4-7  
464 EventSelect 0F6h: HyperTransport™ Link 0 Transmit Bandwidth  
464 EventSelect 0F7h: HyperTransport™ Link 1 Transmit Bandwidth  
464 EventSelect 0F8h: HyperTransport™ Link 2 Transmit Bandwidth  
464 EventSelect 1F9h: HyperTransport™ Link 3 Transmit Bandwidth  
464 EventSelect 4E0h: Read Request to L3 Cache  
465 EventSelect 4E1h: L3 Cache Misses  
466 EventSelect 4E2h: L3 Fills caused by L2 Evictions  
467 EventSelect 4E3h: L3 Evictions  
467 EventSelect 4EDh: Non-cancelled L3 Read Requests