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

Table of Contents

List of Figures

List of Tables

Revision History

Revision 3.00 To Revision 3.06 Changes

- Updated MSRC001_0070[NbDid].
- Clarified 2.4.4 [ACPI Suspend to RAM State (S3)] on page 52.
- Clarified 2.8.8.5 [DRAM Device Initialization] on page 79.
- Clarified 2.8.8.5.1 [Software DDR2 Device Initialization] on page 80.
- Clarified 2.8.8.5.2 [Software DDR3 Device Initialization] on page 82.
- Updated 2.8.8.7 [DRAM Channel Frequency Change] on page 85.
- Corrected 2.9.3.1.2 [Determining The Access Destination for CPU Accesses] on page 109.
- Clarified F2x[1, 0]7C[EnDramInit].
- Clarified F2x[1, 0]9C_x04 programming requirements.
- Clarified F3x44[DisPciCfgCpuErrRsp].
- Clarified F3x180[DisPciCfgCpuMstAbtRsp].
- Clarified 2.4.2.8.2 [Mixed Frequency and Power P-State Configuration Rules] on page 41.
- Updated 2.4.2.8.3 [Mixed Frequency and Power P-State Configuration Sequence] on page 42.
- Updated 2.4.2.11.1 [CPU Core Maximum P-State Transition Sequence After Warm Reset] on page 49.
- Updated 2.4.2.11.2 [CPU Core Minimum P-State Transition Sequence After Warm Reset] on page 50.
- Clarified 2.6.6 [Memory Scrubbers] on page 58.
- Added 2.13.1.5 [Scrub Rate Recommendations] on page 119.
- Clarified 2.13.2 [DRAM Considerations for ECC] on page 120.
- Clarified F3x40[UECC, CECC].
- Clarified F3x4C.
- Clarified F3x58.
- Clarified MSRC001_0015[McStatusWrEn].
- Updated MSRC001_1023 reset values.

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) up to four independent central processing unit cores (referred to as *cores*), (b) up to four 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) up to two 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] on page 285).

1.1 Intended Audience

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

1.2 Reference Documents

- Advanced Configuration and Power Interface (ACPI) Specification. www.acpi.info.
- AMD64 Architecture Programmer's Manual Volume 1: Application Programming, #24592.
- AMD64 Architecture Programmer's Manual Volume 2: System Programming, #24593.
- AMD64 Architecture Programmer's Manual Volume 3: Instruction-Set Reference, #24594.
- AMD64 Architecture Programmer's Manual Volume 4: 128-Bit Media Instructions, #26568.
- AMD64 Architecture Programmer's Manual Volume 5: 64-Bit Media and x87 Floating-Point Instructions, #26569.
- CPUID Specification, #25481.
- AMD Socket F (1207) Processor Functional Data Sheet, #31118.
- AMD Socket Fr2 (1207) Processor Functional Data Sheet, #41698.
- AMD Socket AM2 Processor Functional Data Sheet, #31117.
- AMD Socket AM2r2 Processor Functional Data Sheet, #41697.
- AMD Socket AM3 Processor Functional Data Sheet, #40778.
- AMD Family 10h Processor Electrical Data Sheet, #40014.
- Revision Guide for AMD Family 10h Processors, #41322
- AMD Voltage Regulator Specification, #40182.
- AMD I/O Virtualization Technology (IOMMU) Specification, #34434.
- HyperTransport™ I/O Link Specification. www.hypertransport.org.
- PCI local bus specification. www.pcisig.org.
- System Management Bus (SMBus) specification. www.smbus.org.
- SBI Temperature Sensor Interface (SB-TSI) Specification, #40821.

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] on page 137; 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:

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

is translated as:

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

1.4 Definitions

- **AC coupled**. Refers to the method used for link termination. See section 2.7.2 [Termination and Compensation] on page 60.
- **AP**. Application processor. See section 2.3 [Processor Initialization] on page 23.
- **APML**. Advanced Platform Management Link. See section 2.13.3 [Sideband Interface (SBI)] on page 123.
- **BCS**. Base configuration space. See section 2.11 [Configuration Space] on page 113.
- **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 section 2.3 [Processor Initialization] on page 23.
- **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 section 2.4 [Power Management] on page 27.
- **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 section 2.8.9 [Memory Interleaving Modes] on page 101.
- **Chipkill ECC**. An error correcting code which can recover from DRAM device failures. See section 2.13.2 [DRAM Considerations for ECC] on page 120.
- **CMP**. Chip multi-processing. Refers to processors that include multiple cores. See section 2.1 [Processor Overview] on page 22.
- **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 section 2.2 [System Overview] on page 22.
- **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 section 2.4.2 [P-states] on page 34.
- **Cold reset**. PWROK is deasserted and RESET_L is asserted. See section 2.3 [Processor Initialization] on page 23.
- **CPU or CPU core**. The instruction execution unit(s) of the processor. See section 2.1 [Processor Overview] on page 22.
- **CpuCoreNum**. Specifies the core number. See section 2.9.2 [CPU Cores and Downcoring] on page 108.
- **CPUID function X**. Refers to the CPUID instruction when EAX is preloaded with X. See section 3.9 [CPUID Instruction Registers] on page 285.
- **CS**. Chip select. See F2x[1, 0][5C:40] [DRAM CS Base Address Registers] on page 168.
- **DC coupled**. Refers to the method used for link termination. See section 2.7.2 [Termination and Compensation] on page 60.
- **DCT**. DRAM controller. See section 2.8 [DRAM Controllers (DCTs)] on page 64.
- **DEV**. DMA exclusion vector. See section 2.6.3 [DMA Exclusion Vectors (DEV)] on page 54.
- **DID**. Divisor identifier. Specifies the post-PLL divisor used to reduce the COF. See section 2.4.2 [P-states] on page 34.
- **Display refresh**. Traffic used for display refresh in UMA systems. See section 2.6.4.2.4 [F0x[5C:40]Display Refresh And IFCM] on page 57.
- **Doubleword**. A 32-bit value.
- **Downcoring**. Removal of cores. See section 2.9.2 [CPU Cores and Downcoring] on page 108.
- **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 section 2.8 [DRAM Controllers (DCTs)] on page 64.

- **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] on page 28.
- **DW or DWORD**. Doubleword. A 32-bit value.
- **ECS**. Extended configuration space. See section 2.11 [Configuration Space] on page 113.
- **EDS**. Electrical data sheet. See section 1.2 [Reference Documents] on page 13.
- **FDS**. Functional data sheet; there is one FDS for each package type. See section 1.2 [Reference Documents] on page 13.
- **FID**. Frequency identifier. Specifies the PLL frequency multiplier for a given clock domain. See section 2.4.2 [P-states] on page 34.
- **Ganged**. A link, memory channel, or voltage regulator in which all portions are controlled as one.
- **GB or Gbyte**. Gigabyte; 1,073,741,824 bytes.
- **Gen1**. Refers to older revisions of the link specification and, in particular, link data rates from 0.4 to 2.0 GT/s. See section 2.7 [Links] on page 59.
- **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 section 2.7 [Links] on page 59.
- **#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 section 2.10.3.1 [PROCHOT_L and Hardware Thermal Control (HTC)] on page 112.
- **HTC-active state**. Hardware-controlled lower-power, lower-performance state used to reduce temperature. See section 2.10.3.1 [PROCHOT_L and Hardware Thermal Control (HTC)] on page 112.
- **I2C**. Protocol on which the SVI interface timing is based. See section 2.4.1 [Processor Power Planes And Voltage Control] on page 28, and section 1.2 [Reference Documents] on page 13.
- **IBS**. Instruction based sampling. See section 2.17.2 [Instruction Based Sampling (IBS)] on page 136.
- **IFCM**. Isochronous flow-control mode, as defined in the *HyperTransport™ I/O Link Specification*. See section 2.6.4.2.4 [F0x[5C:40]Display Refresh And IFCM] on page 57.
- **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].
- **IO configuration**. Access to configuration space through IO ports CF8h and CFCh. See section 2.11 [Configuration Space] on page 113.
- **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.
- **KB or Kbyte**. Kilobyte; 1024 bytes.
- **L1 caches**. The level 1 caches of the core including the instruction cache and the data cache.
- **L2 cache**. The level 2 cache of each core.
- **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 section 2.8 [DRAM Controllers (DCTs)] on page 64.
- **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.
- **MB or Mbyte**. Megabyte; 1024 Kbytes.
- **MCT**. Memory controller. See section 2.6.1 [Northbridge (NB) Architecture] on page 54.
- **MCQ**. Memory controller queue. See section 2.6.1 [Northbridge (NB) Architecture] on page 54.
- **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 section 2.11 [Configuration Space] on page 113.
- **MOF**. Maximum operating frequency of the core(s). Normally this is the core COF in P-state 0. See section 2.4.2 [P-states] on page 34.
- **MSR**. Model specific register. The CPU includes several MSRs for general configuration and control. See section 3.10 [MSRs - MSR0000_xxxx] on page 293 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 section 2.1 [Processor Overview] on page 22.
- **NBC**. Node Base Core. The lowest numbered core in the node.
- **NCLK**. The main Northbridge clock. The NCLK frequency is the NB COF.
- **Node ID**. The identifier assigned to each node, [The Node ID Register] F0x60[NodeId].
- **Node**. See section 2.1 [Processor Overview] on page 22.
- **Normalized address**. Addresses used by DCTs. See section 2.6.1 [Northbridge (NB) Architecture] on page 54.
- **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 section 2.4 [Power Management] on page 27.
- **PDS**. Product data sheet. See section 1.2 [Reference Documents] on page 13.
- **PRBS**. Pseudo-random bit sequence.
- **Processor**. See section 2.1 [Processor Overview] on page 22.
- **P-state**. Performance state. See section 2.4 [Power Management] on page 27.
- **PTE**. Page table entry.

- **PVI**. Parallel VID interface. See section 2.4.1 [Processor Power Planes And Voltage Control] on page 28.
- **Quadword**. A 64-bit value.
- **RAS**. Reliability, availability and serviceability (industry term). See section 2.13 [. RAS and Advanced Server Features] on page 114.
- **RX**. Receiver.
- **SBI**. Sideband Interface. Also referred to as APML. See section 2.13.3 [Sideband Interface (SBI)] on page 123.
- **Scrubber**. Background memory checking logic. See section 2.6.6 [Memory Scrubbers] on page 58.
- **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] on page 28.
- **Slam**. Refers to change the voltage to a new value in one step (as opposed to stepping). See section 2.4.1.7 [Hardware-Initiated Voltage Transitions] on page 32.
- **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 section 2.4.1 [Processor Power Planes And Voltage Control] on page 28, 2.13.3 [Sideband Interface (SBI)] on page 123, and section 1.2 [Reference Documents] on page 13.
- **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 section 2.14.2.1 [SMM Overview] on page 127.
- **SMM**. System management mode. See section 2.14.2 [System Management Mode (SMM)] on page 127.
- **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.
- **STC**. Software thermal control. See section 2.10.3.2 [Software Thermal Control (STC)] on page 112.
- **STC-active state**. Software-controlled lower-power, lower-performance state used to reduce temperature. See section 2.10.3.2 [Software Thermal Control (STC)] on page 112.
- **STC thermal zone**. Temperature range which may be programmed to generate interrupts and special bus cycles. See section 2.10.3.2 [Software Thermal Control (STC)] on page 112.
- **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 section 2.7 [Links] on page 59.
- **SVI**. Serial VID interface. See section 2.4.1 [Processor Power Planes And Voltage Control] on page 28.
- **SVM**. secure virtual machine. See section 2.15 [Secure Virtual Machine Mode (SVM)] on page 135.
- **Sync flood**. The propagation of continuous sync packets to all links. See the link specification.
- **TCB**. Trace capture buffer.
- **TCC**. Temperature calculation circuit. See section 2.10 [Thermal Functions] on page 111.
- **Tctl**. Processor temperature control value. See section 2.10.3 [Temperature-Driven Logic] on page 112.
- **TDP**. Thermal design power.
- **Thermal diode**. A diode connected to the THERMDA and THERMDC pins used for thermal measurements. See section 2.10.2 [Thermal Diode] on page 111.
- **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.
- **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.
- **usec**. Microsecond.
- **VDD**. Main power supply to the processor core logic.
- **VDDNB**. Main power supply to the processor NB logic.
- **VID**. Voltage level identifier. See section 2.4.1 [Processor Power Planes And Voltage Control] on page 28.
- **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 section 2.3 [Processor Initialization] on page 23.
- **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 section 2.6.1 [Northbridge (NB) Architecture] on page 54.

1.5 Changes Between Revisions and Product Variations

1.5.1 Major Changes Relative to Family 0Fh Processors

- CPU core additions:
  - Support for up to 4 cores 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 modes.
  - 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.
- 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.
  - 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 APML.
  - 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 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.
- Power management:
  - Simple “fire and forget” operating system interface for P-state changes.
  - Separate core and Northbridge power and clock planes.
  - Support for up to 5 independent P-states for each core.
  - Support for P-state limits controlled by sideband interface (SBI), thermal limits (HTC), or host software; used to limit the P-state requested by the operating system in order to reduce power.

]

1.5.2 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 1: Supported features

Feature	First Revision Supported	First Revision Removed
Unbuffered DDR2 DIMM interface up to 800 MT/s	B	
Unbuffered DDR2 DIMM interface up to 1067 MT/s	B	
Registered DDR2 DIMM interface up to 800 MT/s	B	
Gen3 link and retry DC-coupled mode (coherent links)	B	
Gen3 link and retry DC-coupled mode (non-coherent links)	B	
Link unganging support		
Narrow (2-bit and 4-bit) link		B
CRC insertion on Gen3 coherent links		
Thermal diode (THERMDA and THERMDC)	B	
SMBus-based sideband thermal sensor interface (SB-TSI)	B	
Even traffic distribution across multiple links connected between the same 2 nodes	B	
PVI regulator interface	B	
SVI regulator interface	B	
Single-plane compatible	B	
Dual-plane compatible	B	
Triple-plane compatible		
C1E	B	
Thermal clock throttling (SMC controlled)	B[1]	

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 node. A *node*, is an integrated circuit device that includes (1) one to four 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).

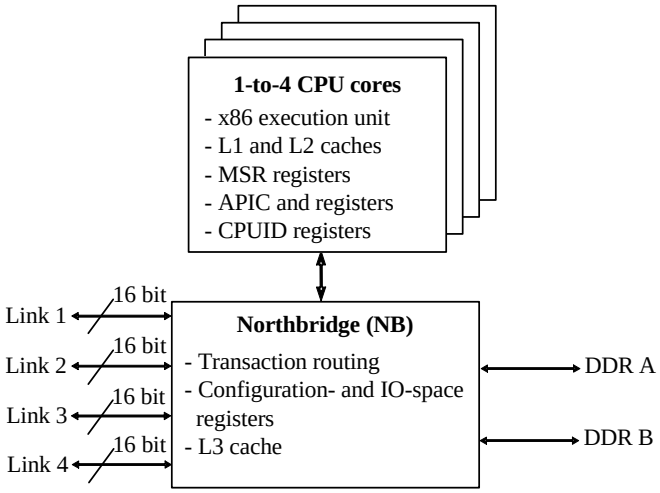

Figure 1: A processor.

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.

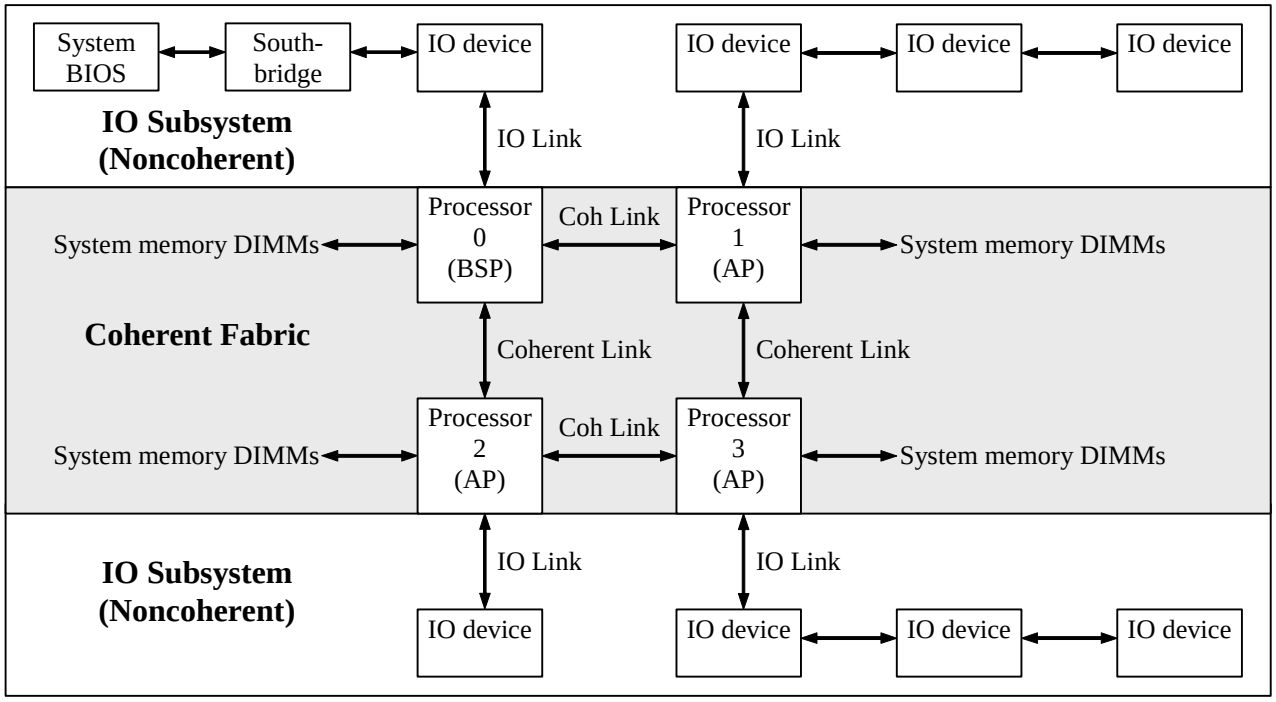

Figure 2: System diagram.

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 (MSR0000_0401, MSR0000_0405, MSR0000_0409, MSR0000_040D, MSR0000_0411, MSR0000_0415). If this is a warm reset then BIOS may check for valid MCA errors and if present save the status for later use (see 2.13.1.3 [Handling Machine Check Exceptions] on page 118).
- 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] on page 24.
- Setup of APIC (2.9.5.1 [ApicId Enumeration Requirements] on page 110).
- Perform coherent link enumeration (routing table and NodeID), as described in 2.6.4.2 [HyperTransport™

Technology Routing] on page 56.
- Configure all I/O-link devices.
  - Set configuration-base and -limit ([The Configuration Map Registers] F1x[EC:E0] [BusNumBase], [BusNumLimit]) and assign BUID.
  - Device enumeration for all I/O-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] on page 27).
- If supported, allow other cores to beginning 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 a an INIT then BIOS vectors away from the cold/warm reset initialization path.
- Determine the history of this reset using the [The Link Initialization Control Register] F0x6C [ColdRstDet] bit. If this is a cold reset then BIOS must clear the [MCi_STATUS] MSRs (MSR0000_0401, MSR0000_0405, MSR0000_0409, MSR0000_040D, MSR0000_0411, MSR0000_0415). If this is a warm reset then BIOS may check for valid MCA errors and if present save the status for use later (see 2.13.1.3 [Handling Machine Check Exceptions] on page 118).
- Setup of local APIC (2.9.5.1 [ApicId Enumeration Requirements] on page 110).
- Configure processor power management (see 2.4 [Power Management] on page 27).

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 CPU core as general 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 CPU core has its own L2 cache.
- The L2 size, L2 associativity, and L2 line size is determined by reading CPUID Fn8000_0006_ECX[L2Size, L2Assoc, L2LineSize]. (Note that 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.
  - For each of the following values of L2Size, the following values are defined:
    - L2Size=128KB: L2Tag=PhysAddr[39:13], L2WayIndex=PhysAddr[12:6].
    - L2Size=256KB: L2Tag=PhysAddr[39:14], L2WayIndex=PhysAddr[13:6].
    - 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, will:
    - Pick an invalid way before picking a valid way.
    - Prioritize the picking of invalid ways such that way 0 is the highest priority and L2WayNum-1 is the lowest priority.
- In order to prevent victimizing L2 data, no more than L2WayNum cache lines may have the same L2WayIndex.
  - Software does not need to know which ways the L2WayNum lines are allocated to for any given value of L2WayIndex, only that invalid ways will be selected for allocation before valid ways will be selected for allocation.

It is recommended that BIOS:

- Assume a simpler allocation of L2 cache memory, being L2WayNum size-aligned blocks of memory, each being L2Size/L2WayNum bytes.
- Assume the minimum L2Size for all configurations.

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 a location in the L2 cache, mapped as write-back DRAM, with 1 read to at least 1 byte of the L2LineSize cache line. 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.
  - Coherency between CPU cores is supported for the write-back DRAM memory type.

Performance monitor event EventSelect 07Fh [L2 Fill/Writeback] on page 348, sub-event bit 1, titled “L2 Writebacks to system“, can be used to indicate whether L2 dirty data was victimized and sent to the disabled memory controller.

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

- Paging must be disabled.
- MSRC001_1022[DIS_SMC_CHK_BUF]=1.
- MSRC001_1022[DIS_HW_PF]=1.
- MSRC001_102A[ClLinesToNbDis]=1.
- MSRC001_0015[INVD_WBINVD]=0.
- CLFLUSH, INVD, and WBINVD must not be used.
- 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_1022[DIS_SMC_CHK_BUF]=0, MSRC001_1022[DIS_HW_PF]=0, MSRC001_102A[ClLinesToNbDis]=0,

MSRC001_0015[INVD_WBINVD].

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 2.4.2.8 [Mixed-Frequency and Power P-State Configuration] on page 40.

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 System 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, 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 3. The maximum number of hops between any two nodes is 3. Table 2 shows the SLIT table entries.

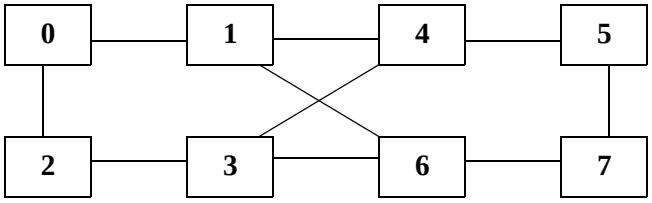

Figure 3: Example 8 node system in twisted ladder topology

Table 2: SLIT table example

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

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 3 provides a summary of each power management state and indicates whether it is supported. Refer to F3x[84:80] [ACPI Power State Control Registers] on page 232 for processor configuration settings for power management.

Table 3: Power management support

ACPI/Power Management State	Supported[1]	Description
G0/S0/C0: Working	Yes	
G0/S0/C0: Core P-state transitions under OS control	Yes	2.4.2.1 [Core P-states] on page 34
G0/S0/C0: Hardware thermal control (HTC)	Yes	2.10.3.1 [PROCHOT_L and Hardware Thermal Control (HTC)] on page 112
G0/S0/C0: Thermal clock throttling (SMC controlled)	Yes[2]	
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)	No	
G0/S0/C1E: Stop-grant Caches not snoopable using MSRC001_0055[C1eOnCmpHlt] (single and multi-core devices)	No	
G0/S0/C1E: Stop-grant Caches not snoopable using MSRC001_0055[SmiOnCmpHlt] (single and multi-core devices)	Yes	2.4.3.1 [C1 Enhanced State (C1E)] on page 51
G1/S1: Stand By (Powered On Suspend)	Yes	
G1/S3: Stand By (Suspend to RAM)	Yes	2.4.4 [ACPI Suspend to RAM State (S3)] on page 52
G1/S4, S5: Hibernate (Suspend to Disk), Shut Down (Soft Off)	Yes	
G3 Mechanical Off	Yes	

1. 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.
2. AMD recommends using PROCHOT_L for thermal throttling and not implementing stop clock based throttling.
3.

2.4.1 Processor Power Planes And Voltage Control

The processor includes the following power planes:
- VDDIO: used for the DRAM and miscellaneous pins on DDR products only. Voltage level is nominally 1.8V or 1.9V in support of DDR2; 1.5V in support of DDR3. This plane is powered during S3 (suspend to RAM).
- VTT: used for the DDR DRAM interface. Voltage level is specified to be half of the VDDIO level. This plane is powered during S3 (suspend to RAM). See section 2.4.4 [ACPI Suspend to RAM State (S3)] on page 52.
- VLDT: used for each of the links. Voltage level is nominally 1.2V.
- VDDA: filtered PLL supply. Voltage level is nominally 2.5V.
- VDD or VDD[1:0]: main supply for core logic. "VDD" refers generically to the core voltage plane(s). Voltage level is specified by the VID interface.
- VDDNB: main supply for NB logic. Voltage level specified by the VID interface.

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. Refer to the AMD

Design Guide for Voltage Regulator Controllers Accepting Serial VID Codes for details on SVI protocol. See section 1.2 [Reference Documents] on page 13.

The processor supports:
- Single-plane platforms in which all the VDD and VDDNB power planes are connected together on the systemboard and controlled as a single power plane through the PVI (F3xA0[PviMode]=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].

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 section 2.4.1.5.3 [Serial VID (SVI) Encodings] on page 31).
- 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 6), 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 pin when F3xA0[PviMode]=1 and by a bit in the data field of the SVI command when F3xA0[PviMode]=0; in SVI mode, the PSI_L pin is always high. It is enabled through F3xA0[PsiVidEn]. PSI_L is asserted if the processor selects a VID code that is lower than the level specified in F3xA0[PsiVid].

2.4.1.5 VID Encodings

The following sections provide VID encoding to VDD translations. Section 2.4.1.5.1 [Boot VID Encodings] on page 30 defines the VID to VDD translation for both protocols prior to PWROK assertion. Sections 2.4.1.5.2 [Parallel VID Interface (PVI) Encodings] on page 30 and 2.4.1.5.3 [Serial VID (SVI) Encodings] on page 31 define the VID to VDD translation following PWROK assertion and protocol initialization (if required).

2.4.1.5.1 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] on page 29. 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 4: 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

2.4.1.5.2 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 5: 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
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

2.4.1.5.3 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 6: 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

Table 6: SVI and internal VID codes

VID[6:0]	VDD	VID[6:0]	VDD	VID[6:0]	VDD	VID[6:0]	VDD
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
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

2.4.1.6 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] on page 30 for additional details.

2.4.1.7 Hardware-Initiated Voltage Transitions

VDD and VDDNB voltage levels may be transitioned during state changes involving boot, reset, P-state, and stop-grant. In some cases, the voltage is *slammed*; this means that the VID code passed to the voltage regulator changes from the old value to the new value without stepping through intermediate values. In other cases, the voltage is stepped; this means that the VID code is *stepped* one increment at a time and held at each value for a voltage-settling time based on [The Clock Power/Timing Control 1 Register] F3xD8[VSRampTime]. Voltages are transitioned as follows:

First state	Second state	Voltage transition
Voltage off	PWROK assert	Voltages slammed to a factory-specified boot VID (for parallel VID interface) or strapped VID code (for serial VID interface).
PWROK assert	RESET_L deassert	Voltages slammed to a factory-specified start-up VID specified by MSRC001_0071[StartupPstate].
Any P-state	Any P-state	Voltage stepped or slammed (based on F3xA0[SlamVidMode]) to the new P-state VID.
Alternate VID	Any P-state	Voltage stepped or slammed (based on F3xA0[SlamVidMode]) to the value captured from the P-state prior to application of the alternate VID.

2.4.1.8 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] on page 28.

2.4.1.8.1 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.

2.4.1.8.2 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 2.4.1.8.1 [Software-Initiated NB Voltage Transitions] on page 33 to control the single-plane through NbVid.

2.4.1.9 SVI Protocol

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

- Only a 400kHz bus clock is supported.

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 though 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 up to 5 independently-controllable frequency planes: each core (up to 4) and the NB; and up to 2 independently-controllable voltage planes: VDD, and VDDNB. Refer to section 2.4.1 [Processor Power Planes And Voltage Control] on page 28 for voltage plane definitions and section 1.5.2 [Supported Feature Variations] on page 21 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] on page 34 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 the product is intended to support; this is specified as the COF of P-state 0 found in MSRC001_0064 (MSRC001_00[68:64]) and F3xD4[NbFid] (for the NB) after a cold reset.
- VID: voltage ID. Specifies the voltage level for a given domain. Refer to 2.4.1.5 [VID Encodings] on page 30 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 2.4.2.8 [Mixed-Frequency and Power P-State Configuration] on page 40 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 2.4.2.5 [BIOS Requirements for P-State Initialization and Transitions] on page 37 are complete.

2.4.2.1.1 Core P-state Control

Core P-states are dynamically controlled by software and are exposed through ACPI objects (refer to section 2.4.2.9 [ACPI Processor P-State Objects] on page 47). Software requests a core P-state change by writing a 3 bit index corresponding to the desired core P-state number to [The P-State Control Register] MSRC001_0062[PstateCmd] of the appropriate core. E.g. to request P3 for core 0 software would write 011b to core 0's MSRC001_0062[PstateCmd]. Refer to [The P-State [4:0] Registers] MSRC001_00[68:64] for mapping of P-state numbers (and corresponding 3 bit indexes) to P-state registers.

Hardware sequences the frequency and voltage changes necessary to complete a P-state transition as specified by 2.4.2.4 [P-state Transition Behavior] on page 35 with no additional software interaction required. Core P-states are changed without interaction with the external chipset. [The P-State Status Register] MSRC001_0063[CurPstate] reflects the current frequency component (COF) of each core as a 3 bit index corresponding to the current P-state number. E.g. Core 1 MSRC001_0063[CurPstate] = 010b indicates core 1 is at the P2 COF (specified by MSRC001_0066[CpuFid and CpuDid]).

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 1.5.2 [Supported Feature Variations] on page 21 for package/platform specific voltage plane support. E.g. The VID for a 4-core single voltage plane system must be maintained at the highest level required for all 5 frequency domains (4 cores and NB). Refer to section 2.4.2.4 [P-state Transition Behavior] on page 35 for details on hardware P-state voltage control. Section 2.4.1.6 [BIOS Requirements for Power Plane Initialization] on page 32 specifies the processor initialization requirements for voltage plane control.

2.4.2.2 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] and [The Software Thermal Control (STC) Register] F3x68[StcPstateLimit]. 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[PslApicLoEn and PslApicHiEn]. In addition, the maximum value P-state, regardless of the source, is limited as specified in MSRC001_0061[PstateMaxVal].

2.4.2.3 P-state Bandwidth Requirements

- The frequency relationship of (core COF / NB COF) <= 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) <= 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 >= 400Mhz.

2.4.2.4 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 section 2.4.1.7 [Hardware-Initiated Voltage Transitions] on page 32.

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. 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.
- 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).
- If a third command is issued placing core 1 back into P0, then:
  - 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 it's 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 it's 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 2.4.2.11 [BIOS COF and VID Requirements After Warm Reset] on page 49.
  - 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 also section 2.4.1.8 [Software-Initiated Voltage Transitions] on page 33.
- 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] and [The Software Thermal Control (STC) Register] F3x68[StcPstateLimit]. 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.

2.4.2.5 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] on page 232 according to the settings in Table 54.
2. Configure the Northbridge COF and VID for each processor appropriately based on the sequence described in 2.4.2.6 [BIOS Northbridge COF and VID Configuration] on page 37.
3. Complete the 2.4.1.6 [BIOS Requirements for Power Plane Initialization] on page 32.
4. Complete the 2.4.2.7 [Processor-Systemboard Power Delivery Compatibility Check] on page 39.
5. Determine the valid set of P-states:
   - Based on the sequence described in 2.4.2.8 [Mixed-Frequency and Power P-State Configuration] on page 40 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.
6. 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 2.4.2.9 [ACPI Processor P-State Objects] on page 47. 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[PllLockTime].
- Configure F3xD4[NbClkDivApplyAll, NbClkDiv, and ClkRampHystSel].

2.4.2.6 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 {
- 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].

}

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

If F3x1FC[NbCofVidUpdate]=0 for all processors in the system 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 2.4.2.6 [BIOS Northbridge COF and VID Configuration] on page 37.

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. Copy NewNbVid to MSRC001_0064[NbVid] for all cores on the local processor.
3. Request a transition to P1 (Write MSRC001_0062[PstateCmd]=001b) for all cores on the local processor.
4. Request a transition to P0 (Write MSRC001_0062[PstateCmd]=000b) on core 0 of the local processor.
5. Wait for MSRC001_0063[CurPstate]=000b on core 0 of the local processor.
6. Copy NewNbFid to F3xD4[NbFid] and set F3xD4[NbFidEn] on the local processor.
7. Repeat steps 1 through 6 for each processor in the system.
8. 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].
9. 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][PstateEn]=1.
   - If F3x1FC[NbVidUpdateAll]=1 copy NewNbVid to MSRC001_00[68:64][NbVid] where MSRC001_00[68:64][PstateEn]=1.
10. For each processor in the system, transition all cores to MSRC001_0071[StartupPstate] using MSRC001_0062[PstateCmd].

2.4.2.6.2 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[StartupPstate] 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. Request a transition to P1 (Write MSRC001_0062[PstateCmd]=001b) for all cores on the local processor.
7. Request a transition to P0 (Write MSRC001_0062[PstateCmd]=000b) on core 0 of the local processor.
8. Wait for MSRC001_0063[CurPstate]=000b on core 0 of the local processor.
9. Copy NewNbFid to F3xD4[NbFid] and set F3xD4[NbFidEn] on the local processor.
10. Repeat steps 4 through 9 for each processor in the system.
11. Latch the VID code driven on the processor VID[5:0] lines and preserve the value to the VDDNB regulator.
12. Request a transition to P1 (Write MSRC001_0062[PstateCmd]=001b) on core 0 of each processor.

13. Expose the VDD regulator inputs to the VID code driven on the processor VID[5:0] lines while leaving the VDDNB regulator input latched.
14. 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].
15. Copy MSRC001_00[68:64][CpuVid] to MSRC001_00[68:64][NbVid] for every core in the system.
16. For each processor in the system, transition all cores to MSRC001_0071[StartupPstate] using MSRC001_0062[PstateCmd].

2.4.2.7 Processor-Systemboard Power Delivery Compatibility Check

BIOS must disable processor P-states that require higher power delivery than the systemboard can support. This power delivery compatibility check is designed to prevent system failures caused by exceeding the power delivery capability of the systemboard for the power plane(s) that contain the core(s). Refer to section 2.4.1 [Processor Power Planes And Voltage Control] on page 28 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 processor node. Note that 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.

Dual-plane systems:

```
ProcIddMax = MSRC001_00[68:64][IddValue] * 1/10^MSRC001_00[68:64][IddDiv] *
   (F3xE8[CmpCap]+1)- F3x1FC[SinglePlaneNbIdd] * 2;
```

Single-plane systems:

The power delivery check should be applied starting with P0 and continue with increasing P-state indexes (1, 2, 3, and 4) for all enabled P-states. Once a compatible P-state is found using the ProcIddMax equation the check is complete. All processor P-states with higher indexes are defined to be lower power and performance, and are therefore compatible with the systemboard.

Single power plane P1 example:
- MSRC001_0065[IddValue] = 32d
- MSRC001_0065[IddDiv] = 0d
- F3xE8[CmpCap] = 1d
- ProcIddMax = 32 * 1 * 2 = 64 A per plane

The systemboard must be able to supply >= 64 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 A per plane then BIOS must set MSRC001_0065[PstateEn]=0 for all cores on this processor node, and continue by checking P2 in the same fashion.

Dual power plane P1 example:
- MSRC001_0065[IddValue] = 32d
- MSRC001_0065[IddDiv] = 0d
- F3xE8[CmpCap] = 1d
- F3x1FC[SinglePlaneNbIdd] = 10d

- ProcIddMax = 32 * 1 * 2 - 20 = 44 A per plane

The systemboard must be able to supply >= 44 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 A per plane then BIOS must set MSRC001_0065[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 MSRC001_0063[CurPstate] is disabled by the power delivery compatibility check, then BIOS must request a transition to an enabled P-state using MSRC001_0062[PstateCmd] and wait for MSRC001_0063[CurPstate] to reflect the new value.
2. Copy the contents of the enabled P-state 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 (PstateEn=0).
3. Request a P-state transition to the P-state MSR containing the COF/VID values currently applied. E.g. If MSRC001_0063[CurPstate]=100b and P4 P-state MSR information is copied to P2 in step 2, then BIOS should write 010b to MSRC001_0062[PstateCmd] and wait for MSRC001_0063[CurPstate] to reflect the new value.
4. Adjust the following P-state parameters affected by the P-state MSR copy by subtracting the number of P-states that are disabled by the power delivery compatibility check. This calculation should not wrap, but saturate at 0. E.g. if P0 and P1 are disabled, then each of the following register fields should have 2 subtracted from them:
   - F3x64[HtcPstateLimit]
   - F3x68[StcPstateLimit]
   - F3xDC[PstateMaxVal]

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

1. If MSRC001_0063[CurPstate]!=F3xDC[PstateMaxVal], then write F3xDC[PstateMaxVal] to MSRC001_0062[PstateCmd] and wait for MSRC001_0063[CurPstate] to reflect the new value.
2. If F3xDC[PstateMaxVal]!=000b copy the contents of the P-state MSR pointed to by F3xDC[PstateMaxVal] to MSRC001_0064 and set MSRC001_0064[PstateEn]; Write 000b to MSRC001_0062[PstateCmd] and wait for MSRC001_0063[CurPstate] to reflect the new value.
3. Adjust the following fields to 000b.
   - F3x64[HtcPstateLimit]
   - F3x68[StcPstateLimit]
   - F3xDC[PstateMaxVal]

2.4.2.8 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 2.4.2.9.2 [_PSS (Performance Supported States)] on

page 47.

- 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 2.4.2.8.2 [Mixed Frequency and Power P-State Configuration Rules] on page 41 and 2.4.2.8.3 [Mixed Frequency and Power P-State Configuration Sequence] on page 42 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 2.4.2.8.1 [Mixed Power P-State Configuration Sequence] on page 41 defines the required BIOS modifications to [The P-State [4:0] Registers] MSRC001_00[68:64].

2.4.2.8.1 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 2.4.2.9.2 [_PSS (Performance Supported States)] on page 47.
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.

2.4.2.8.2 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 2.4.2.7 [Processor-Systemboard Power Delivery Compatibility Check] on page 39), 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 2.4.2.8.3 [Mixed Frequency and Power P-State Configuration Sequence] on page 42. 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 fre-

quency P-state configuration sequence for any P-state.
- No P-state changes are allowed until all appropriate steps of the sequence are complete.

2.4.2.8.3 Mixed Frequency and Power P-State Configuration Sequence

1. Verify the rules in section 2.4.2.8.2 [Mixed Frequency and Power P-State Configuration Rules] on page 41 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 7: 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 ***100 W***	***2.5 GHz*** ***100 W***	***2.5 GHz*** 100 W	2.5 GHz ***100 W***
P1	2.2 GHz[1] 80 W	2.4 GHz 80 W	3.0 GHz[1] 90 W	2.3 GHz[1] 60 W	2.2 GHz[1] 80 W	2.4 GHz 80 W	3.0 GHz[1] 90 W	2.3 GHz[1] 60 W
P2	1.8 GHz 70 W	2.2 GHz[1] 70 W	2.4 GHz 80 W	1.8 GHz 50 W	1.8 GHz 70 W	2.2 GHz[1] 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[2] 40 W	1.2 GHz 60 W	1.0 GHz 60 W	1.6 GHz 70 W	500 MHz[2] 40 W
P4	500 MHz[2] 50 W	600 MHz[2] 50 W	500 MHz[2] 60 W	N/A[3]	500 MHz[2] 50 W	600 MHz[2] 50 W	500 MHz[2] 60 W	N/A[3]

Refer to Table 12 for notes.

3. 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 8: 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 *100 W*	*2.5 GHz* *100 W*	*2.5 GHz* 100 W	2.5 GHz *100 W*
P1	2.2 GHz[1] 80 W	2.4 GHz 80 W	3.0 GHz[1] 90 W	2.3 GHz[1] 60 W	2.2 GHz[1] ***90 W***	2.4 GHz 80 W	***2.2 GHz***[1] 90 W	***2.2 GHz***[1] ***90 W***
P2	1.8 GHz 70 W	2.2 GHz[1] 70 W	2.4 GHz 80 W	1.8 GHz 50 W	1.8 GHz 70 W	2.2 GHz[1] ***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[2] 40 W	1.2 GHz 60 W	1.0 GHz 60 W	1.6 GHz 70 W	500 MHz[2] 40 W
P4	500 MHz[2] 50 W	600 MHz[2] 50 W	500 MHz[2] 60 W	N/A[3]	500 MHz[2] 50 W	600 MHz[2] 50 W	500 MH[2]z 60 W	N/A[3]

Refer to Table 12 for notes.

4. 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 9: 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[1] 80 W	2.4 GHz 80 W	3.0 GHz[1] 90 W	2.3 GHz[1] 60 W	2.2 GHz[1] *90 W*	2.4 GHz 80 W	*2.2 GHz*[1] 90 W	*2.2 GHz*[1] *90 W*
P2	1.8 GHz 70 W	2.2 GHz[1] 70 W	2.4 GHz 80 W	1.8 GHz 50 W	1.8 GHz 70 W	2.2 GHz[1] *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[2] 40 W	1.2 GHz 60 W	1.0 GHz 60 W	1.6 GHz 70 W	500 MHz[2] ***60 W***
P4	500 MHz[2] 50 W	600 MHz[2] 50 W	500 MHz[2] 60 W	N/A[3]	500 MHz[2] ***60 W***	***500 MHz***[2] ***60 W***	500 MHz[2] 60 W	N/A[3]

Refer to Table 12 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 10: 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[1] 80 W	2.4 GHz 80 W	3.0 GHz[1] 90 W	2.3 GHz[1] 60 W	2.2 GHz[1] *90 W*	***N/A***[3]	*2.2 GHz*[1] 90 W	*2.2 GHz*[1] *90 W*
P2	1.8 GHz 70 W	2.2 GHz[1] 70 W	2.4 GHz 80 W	1.8 GHz 50 W	1.8 GHz 70 W	2.2 GHz[1] *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[2] 40 W	1.2 GHz 60 W	1.0 GHz 60 W	1.6 GHz 70 W	500 MHz[2] *60 W*
P4	500 MHz[2] 50 W	600 MHz[2] 50 W	500 MHz[2] 60 W	N/A[3]	500 MHz[2] *60 W*	*500 MHz*[2] *60 W*	500 MHz[2] 60 W	N/A[3]

Refer to Table 12 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 11: 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[1] 80 W	2.4 GHz 80 W	3.0 GHz[1] 90 W	2.3 GHz[1] 60 W	2.2 GHz[1] *90 W*	*N/A*[3]	*2.2 GHz*[1] 90 W	*2.2 GHz*[1] *90 W*
P2	1.8 GHz 70 W	2.2 GHz[1] 70 W	2.4 GHz 80 W	1.8 GHz 50 W	***N/A***[3]	2.2 GHz[1] *90 W*	***N/A***[3]	***1.0 GHz*** ***70 W***
P3	1.2 GHz 60 W	1.0 GHz 60 W	1.6 GHz 70 W	500 MHz[2] 40 W	***1.0 GHz*** ***70 W***	1.0 GHz ***70 W***	***1.0 GHz*** 70 W	500 MHz[2] *60 W*
P4	500 MHz[2] 50 W	600 MHz[2] 50 W	500 MHz[2] 60 W	N/A[3]	500 MHz[2] *60 W*	*500 MHz*[2] *60 W*	500 MHz[2] 60 W	N/A[3]

Refer to Table 12 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 12: 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 *100 W*	*2.5 GHz* *100 W*	*2.5 GHz* 100 W	2.5 GHz *100 W*
P1	2.2 GHz[1] 80 W	2.4 GHz 80 W	3.0 GHz[1] 90 W	2.3 GHz[1] 60 W	2.2 GHz[1] *90 W*	*N/A*[3]	*2.2 GHz*[1] 90 W	*2.2 GHz*[1] *90 W*
P2	1.8 GHz 70 W	2.2 GHz[1] 70 W	2.4 GHz 80 W	1.8 GHz 50 W	*N/A*[3]	2.2 GHz[1] *90 W*	*N/A*[3]	*1.0 GHz* *70 W*
P3	1.2 GHz 60 W	1.0 GHz 60 W	1.6 GHz 70 W	500 MHz[2] 40 W	*1.0 GHz* *70 W*	1.0 GHz *70 W*	*1.0 GHz* 70 W	500 MHz[2] *60 W*
P4	500 MHz[2] 50 W	600 MHz[2] 50 W	500 MHz[2] 60 W	N/A[3]	500 MHz[2] *60 W*	*500 MHz*[2] *60 W*	500 MHz[2] 60 W	N/A[3]

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 13: 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 2.4.2.9.2 [_PSS (Performance Supported States)] on page 47 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 2.4.2.9.2 [_PSS (Performance Supported States)] on page 47 for details on _PSS object creation.

2.4.2.9 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 2.4.2.7 [Processor-Systemboard Power Delivery Compatibility Check] on page 39), no ACPI objects should be generated.

2.4.2.9.1 _PCT (Performance Control)

BIOS must declare the performance control object parameters as functional fixed hardware. This definition indicates the processor driver understands the architectural definition of the P-state interface associated with CPUID Fn8000_0007[HwPstate]=1.

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

2.4.2.9.2 _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] on page 34 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 2.4.1.5 [VID Encodings] on page 30
  - Power(mW) = voltage * MSRC001_00[68:64][IddValue] * 1/10^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: TransitionLatency = BusMasterLatency = (15 steps * F3xD4[PowerStepDown] ns/step / 1000 us/ns) + (15 steps * F3xD4[PowerStepUp] ns/step / 1000 us/ns)
  - If MSRC001_00[68:64][CpuFid] is different for any P-states where MSRC001_00[68:64][PstateEn]=1: TransitionLatency = BusMasterLatency = (15 steps * F3xD4[PowerStepDown] ns/step / 1000 us/ns) + F3xA0[PllLockTime] us + (15 steps * F3xD4[PowerStepUp] ns/step / 1000 us/ns)

    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[PllLockTime] = 011b (4 us)
    TransitionLatency = BusMasterLatency = (15 steps * 50 ns/step / 1000 us/ns) + 4 us + (15 steps * 50 ns/step / 1000 us/ns) = 5.5 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

2.4.2.9.3 _PPC (Performance Present Capabilities)

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

2.4.2.9.4 _PSD (P-State Dependency)

The ACPI 3.0 _PSD object is required be generated for each core as follows:

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

2.4.2.9.5 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

2.4.2.10 XPSS (Microsoft® Extended PSS) Object

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

2.4.2.11 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 2.4.2.4 [P-state Transition Behavior] on page 35 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 2.4.2.11.1 [CPU Core Maximum P-State Transition Sequence After Warm Reset] on page 49 and 2.4.2.11.2 [CPU Core Minimum P-State Transition Sequence After Warm Reset] on page 50. Transitioning to the minimum P-state after warm reset is recommended to prevent undesired system behavior if a warm reset occurs before the 2.4.2.7 [Processor-Systemboard Power Delivery Compatibility Check] on page 39 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].

2.4.2.11.1 CPU 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 F3xDC[PstateMaxVal] = 0, then exit this sequence (no further steps are executed).
3. If F3xA0[PviMode]=0 transition MSRC001_0070[CpuVid] to MSRC001_0064[CpuVid]; else transition MSRC001_0070[NbVid] to the lowest VID (greatest voltage) of MSRC001_0064[CpuVid] and MSRC001_0064[NbVid]. Refer to section 2.4.1.8 [Software-Initiated Voltage Transitions] on page 33 for additional rules.
4. If MSRC001_0071[CurPstate] != 000b, go to step 17.
5. If MSRC001_0071[CurPstate] = 000b and F3xDC[PstateMaxVal] >= 001b, go to step 15.
6. Copy MSRC001_0064 to MSRC001_0065.
7. Write 001b to F3xDC[PstateMaxVal].
8. Write 001b to MSRC001_0062[PstateCmd].
9. Wait for MSRC001_0071[CurCpuFid] = MSRC001_0065[CpuFid] and MSRC001_0071[CurCpuDid] = MSRC001_0065[CpuDid] (See MSRC001_00[68:64]).
10. Write 000b to MSRC001_0062[PstateCmd].
11. Wait for MSRC001_0071[CurCpuFid] = MSRC001_0064[CpuFid] and MSRC001_0071[CurCpuDid] = MSRC001_0064[CpuDid] (See MSRC001_00[68:64]).
12. If required, transition the NB COF and VID to MSRC001_0064[NbDid, NbVid] using the sequence defined in section 2.4.2.11.3 [NB COF and VID Transition Sequence After Warm Reset] on page 50.
13. Write 0b to MSRC001_0065[PstateEn].
14. Write 000b to F3xDC[PstateMaxVal] and exit the sequence (no further steps are required).
15. Write 001b to MSRC001_0062[PstateCmd].
16. Wait for MSRC001_0071[CurCpuFid] = MSRC001_0065[CpuFid] and MSRC001_0071[CurCpuDid] = MSRC001_0065[CpuDid] (See MSRC001_00[68:64]).
17. Write 000b to MSRC001_0062[PstateCmd].
18. Wait for MSRC001_0071[CurCpuFid] = MSRC001_0064[CpuFid] and MSRC001_0071[CurCpuDid] =

MSRC001_0064[CpuDid] (See MSRC001_00[68:64]).
19. If required, transition the NB COF and VID to MSRC001_0064[NbDid, NbVid] using the sequence defined in section 2.4.2.11.3 [NB COF and VID Transition Sequence After Warm Reset] on page 50.

2.4.2.11.2 CPU 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 F3xDC[PstateMaxVal] = 0, then exit this sequence (no further steps are executed).
3. If MSRC001_0071[CurPstate] != F3xDC[PstateMaxVal], go to step 6.
4. Copy F3xDC[PstateMaxVal] - 1 to MSRC001_0062[PstateCmd].
5. 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.
6. Copy F3xDC[PstateMaxVal] to MSRC001_0062[PstateCmd].
7. 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].
8. If F3xA0[PviMode]=0 transition MSRC001_0070[CpuVid] to CpuVid from [The P-State [4:0] Registers] MSRC001_00[68:64] pointed to by F3xDC[PstateMaxVal]; else transition MSRC001_0070[NbVid] to the lowest VID (highest voltage) of NbVid and CpuVid from [The P-State [4:0] Registers] MSRC001_00[68:64] pointed to by F3xDC[PstateMaxVal]. Refer to section 2.4.1.8 [Software-Initiated Voltage Transitions] on page 33 for additional rules.
9. 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 2.4.2.11.3 [NB COF and VID Transition Sequence After Warm Reset] on page 50.

2.4.2.11.3 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 2.4.1.8 [Software-Initiated Voltage Transitions] on page 33.
2. Write 0 to MSRC001_0070[NbDid].
3. Wait for MSRC001_0071[NbDid]=0.

If the destination NbDid=1:
1. Write 1 to MSRC001_0070[NbDid].
2. Wait for MSRC001_0071[NbDid]=1.
3. Repeat steps 1 and 2 for all cores.
4. Transition MSRC001_0070[NbVid] to the destination NbVid using the sequence defined in section 2.4.1.8 [Software-Initiated Voltage Transitions] on page 33.

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. The actions taken by the processor when a low-power state is entered are defined by [The ACPI Power State Control Registers] F3x[84:80]. C0 and C1 are ACPI-defined states, see the ACPI specification for details. C1E is an AMD specific state.

2.4.3.1 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.
- C1E is only supported on single node systems.

2.4.3.1.1 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 section 2.4.3.1.2.1 [SMM Handler Requirements for C1E] on page 51 for a description.

2.4.3.1.2 BIOS Requirements to Initialize SMI Initiated C1E

On all cores:
- MSRC001_0055[SmiOnCmpHalt] = 1.
- 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.
- BIOS must also setup the SMM handler as described below.

2.4.3.1.2.1 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
```

2.4.4 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. 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 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 DRAM Controller Select High Register] F2x114
   - [The Memory Controller Configuration Low Register] F2x118
   - [The Memory Controller Configuration High Register] F2x11C
   - [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 Miscellaneous Register 2] F2x[1, 0]A8
4. Restore [The DRAM Configuration High Register] F2x[1, 0]94. In unganged mode, follow the frequency initialization procedure specified in 2.8.8.7 [DRAM Channel Frequency Change] on page 85.
5. Wait for F2x[1, 0]94[FreqChgInPrg]=0.
6. Restore 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]. See section 2.8.8.7 for additional requirements.
9. Wait for F2x[1, 0]90[ExitSelfRef]=0.
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[302:301, 202:201, 102:101, 02:01]
    - [The DRAM Write ECC Timing Register] F2x[1, 0]9C_x[303, 203, 103, 03]
    -
    - [The DRAM Read DQS Timing Control [High:Low] Registers] F2x[1, 0]9C_x[306:305, 206:205, 106:105, 06:05]
    - [The DRAM Read DQS ECC Timing Control Register] F2x[1, 0]9C_x[307, 207, 107, 07]
    - [The DRAM Phy Predriver Calibration Register] F2x9C_x0A
    - [The DRAM DQS Receiver Enable Timing Control Registers] F2x[1, 0]9C_x[2B:10]
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[45:30]

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] on page 28 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.

2.5 Processor State Transition Sequences

2.5.1 ACPI Power State Transitions

2.6 The Northbridge (NB)

Each processor 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 section 2.4.1 [Processor Power Planes And Voltage Control] on page 28 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 section 2.9.3 [Access Type Determination] on page 108 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] on page 109) 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[GartTblBaseAddr], 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].

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 4 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.

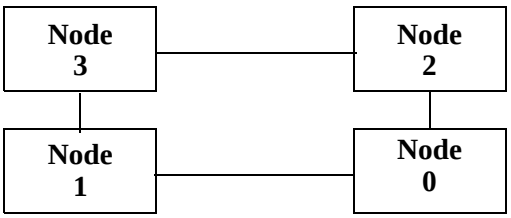

Figure 4: Sample four-node configuration

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 4. 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 to 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 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 the following 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.

2.6.4.2.3 Link Traffic Distribution

Link traffic distribution is a mechanism to reduce coherent link congestion by distributing the traffic over multiple links. It 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 F0x[5C:40]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. 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 also [The Link Base Channel Buffer Count Registers] F0x[F0, D0, B0, 90] for IFCM buffer requirements and [The SRI to XCS Token Count Register] F3x140 for IFCM and display refresh token requirements.

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.

2.6.6 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 (e.g., seconds or minutes), it is a minimum exposure for scrubbing to be suspended during the halt.

For recommendations on scrub rates, see section 2.13.1.5 [Scrub Rate Recommendations] on page 119.

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 5.2 GT/s in the revision 3 specification. Note: 2.4 GT/s and 2.8 GT/s are supported as specified in the revision 3 specification only, not as specified in the revision 2 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, DC-coupled mode; if the two link frequencies are the same, then they may be any supported frequency; if the two link frequencies are different, then they are required to be one of the following ratios to each other: 8:1, 6:1, 4:1, 2:1; legal combinations are {2.4, 0.4}, {4.8, 0.8}, {4.8, 2.4}, {4.8, 1.2}, {4.0, 2.0}, {3.2, 1.6}, {3.2, 0.8}, {3.2, 0.4}, {2.4, 1.2}, {1.6, 0.8}, {1.6, 0.4}, and {0.8, 0.4} GT/s). In unganged mode, neither, either, or both of the two 8-bit sublinks may be left unconnected. In unganged 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.

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

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

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 unganged.
- The link/sublink is connected to another device via DC-coupled termination.
- The link/sublink is not connected with inputs terminated to the proper state to indicate this.
- 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 DC-coupled device is detected on the other side of the link.
- DC-coupled links: link is strapped in the unconnected state per the link specification.

2.7.1.4 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

2.7.2 Termination and Compensation

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

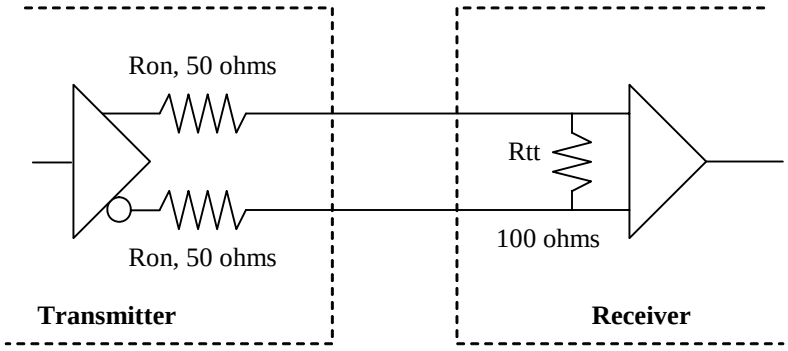

Figure 5: Link DC termination mode.

Ron and Rtt 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 Ron and Rtt 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 nodes NB. The requirements are as follows:

- 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 >= HTCLK.
  - 8-bit or less, ganged links: NCLK >= HTCLK / 2.
  - All unganged links: NCLK >= 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 be 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 14: Link disconnect controls

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

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]).
  - 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 (including link width/frequency changes, S1-based power management, and stutter mode) LDTSTOP_L must not deassert less than 10 microseconds after the processor broadcasts the STOPGRANT message.
- The processor requires a minimum LDTSTOP_L deassertion time of 3 microseconds.
- Note that narrow and slow links and 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.
- Software initiated link width and frequency changes are only supported during link initialization.
- 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 15: Supported link operational modes

Frequency	200-1000MHz	1200-2600MHz	
Coupling/ Link Type	DC	DC non-coherent operational	DC coherent or non-coherent test/debug
Termination	RXDIFF	RXDIFF	RXDIFF
8b10b	No	No	No
Scrambling	No	Yes	Optional
Gen3 Training	No	Yes	Yes
Retry	No	Required	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 receiver 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.

2.8 DRAM Controllers (DCTs)

The DCTs support DDR2 DIMMs or DDR3 DIMMs. Products may be configurable between DDR2 and DDR3 operation.

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

For DDR 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*.

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 section 1.5.2 [Supported Feature Variations] on page 21 for information about supported package/DRAM configurations.

For DDR products, 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 section 2.8.1 [DCT Configuration Registers] on page 65). When the DCTs are in 64-bit mode, a logical DIMM is equivalent to a 64-bit physical DIMM and each channel is controlled by a different DCT.

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.
- The DCTs do not support different DRAM types (DDR2 and DDR3) on the same channel or between channels.
- The DCTs do not support the mixing of unbuffered and registered DIMMs on the same channel or between channels.
- The DCTs do not support the mixing of ECC and non-ECC DIMMs on the same channel or between channels.

Table 16 below list the DIMM speeds supported by the processor for different configurations. See section 2.8.8.4.8 [DRAM Address Timing and Output Driver Compensation Control] on page 75 for detailed information on supported memory bus loads and for configuration settings based on loads.

Table 16: DDR2 Unbuffered and Registered DIMM Support (per channel)

DIMM Slots	DIMMs	DIMMs (by type[1])		Frequency (MT/s)	
		QR	SR or DR	Unbuffered DIMMs on motherboard	Registered DIMMs on motherboard
1	1	0	1	1066	800
		1	0	-	533
2	1	0	1	1066	800
		1	0	-	533
	2	0	2	800	800
		1 or 2	1 or 0	-	667
4	1 or 2	-	any	-	800
4	3 or 4	-	any	-	533

1. SR = Single Rank, DR = Dual Rank, QR = Quad Rank.

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 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.

2.8.3 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.

2.8.4 Ganged or Unganged Mode Considerations

Typical systems built from multi-core products benefit from the additional parallelism generated by using the

two DCTs in unganged mode. Single core products or products that require additional ECC correction capabilities (see 2.13.2 [DRAM Considerations for ECC] on page 120) should implement ganged mode.

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

2.8.5 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] on page 55. 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))<<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))<<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, device, 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, device, func2, 0x114)<<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_FFFFFFC0 -
ChannelOffsetLong & 0x0000FFFF_FF800000;
//Remove Node ID (in case of processor interleaving)
Temp = ChannelAddrLong & 0xFC0;
ChannelAddrLong = (ChannelAddrLong >>Ilog & 0xFFFF_FFFFF000)|Temp;
//Remove Channel interleave and hash
if(DctSelIntLvEn && DctSelHiRngEn==0 && DctGangEn==0)
{
   if(DctSelIntLvAddr & 1 != 1)
      ChannelAddrLong = (ChannelAddrLong>>1) & 0xFFFFFFFF_FFFFFFC0;
   else if(DctSelIntLvAddr == 1)
   {
      Temp = ChannelAddrLong & 0xFC0;
      ChannelAddrLong = ((ChannelAddrLong & 0xFFFFFFFF_FFFFE000) >> 1)| Temp;
   }
   else
   {
      Temp = ChannelAddrLong & 0x1C0;
      ChannelAddrLong = ((ChannelAddrLong & 0xFFFFFFFF_FFFFFC00) >> 1)| Temp;
   }
InputAddr = ChannelAddrLong>>8;
for(CS = 0; CS < 8; CS++)
{
   F2Offset = 0x40 + (CS << 2);
   if ((CS % 2) == 0)
      F2MaskOffset = 0x60 + (CS << 1);
   else
      F2MaskOffset = 0x60 + ((CS-1) << 1);
   if(ChannelSelect)
   {
      F2Offset+=0x100;
      F2MaskOffset+=0x100;
   }
   CSBase = Get_PCI(bus0, dev24 + NodeID, func2, F2Offset);
   CSEn = CSBase & 0x00000001;
   CSBase = CSBase & 0x1FF83FE0;
   CSMask = Get_PCI(bus0, dev24 + NodeID, func2, F2MaskOffset);
   CSMask = (CSMask | 0x0007C01F) & 0x1FFFFFFF;
   if(CSEn && ((InputAddr & ~CSMask) == (CSBase & ~CSMask)))
   {
      CSFound = 1;
      OnlineSpareCTL = Get_PCI(bus0, dev24 + NodeID, func3, 0xB0);
      if(ChannelSelect)
      {
         SwapDone = (OnlineSpareCTL >> 3) & 0x00000001;
         BadDramCs = (OnlineSpareCTL >> 8) & 0x00000007;
         if(SwapDone && CS == BadDramCs) CS=CH1SPARE_RANK;
      }
      else
```

```
                {
                   SwapDone = (OnlineSpareCTL >> 1) & 0x00000001;
                   BadDramCs = (OnlineSpareCTL >> 4) & 0x00000007;
                   if(SwapDone && CS == BadDramCs) CS=CH0SPARE_RANK;
                }
                break;
             }
          }
       }
   }
if(CSFound) break;
}  // for each DramRange
return(CSFound,NodeID,ChannelSelect,CS);
```

2.8.6 DRAM Controller Direct Response Mode

The DCT supportsdirect response mode for responding to a cache line fill request before the DCT is initialized (see section 2.8.8.5).

The target DCT responds to a cache line fill request by returning 64 bytes of all ones without issuing a read transaction on the DRAM bus. The BIOS uses this featureto allocate cache lines for temporary data storage. See section 2.3.3 for more information. The controller exits direct response mode when either F2x[1, 0]7C[EnDramInit] or F2x[1, 0]90[InitDram] is set to 1.

2.8.7 DRAM Data Burst Mapping

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

- In unganged mode, when F2x110[DctDatIntLv] = 0, a 64-byte request is mapped to each of the eight sequential data beats as QW0, QW1...QW7.
- In unganged mode, when F2x110[DctDatIntLv] = 1, the order of cache data to QW on the bus is the same except that even and odd bits are interleaved on the DRAM bus as follows:
  - For every 8 bytes in the cache line, even bits map to QW0, QW2, QW4, and QW6 on the DRAM bus.
  - For every 8 bytes in the cache line, odd bits map to QW1, QW3, QW5, and QW7 on the DRAM bus.
- In unganged mode, a 64-byte request is mapped to each of the eight sequential data beats as QW0, QW1...QW7.
- 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).

2.8.8 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. Phy and controller mode configuration. See section 2.8.8.1.
2. Phy compensation initialization. See section 2.8.8.2.
3. DRAM controller and device initialization. See the note at the end of this section.
   a. Program SPD timings. See section 2.8.8.3.
   b. Program Non-SPD timings. See section 2.8.8.4 and all sub-sections.
   c. DRAM device initialization. See section 2.8.8.5.
4. Phy Fence programmingSee section 2.8.8.6.

5. DRAM Write Leveling ((if using DDR3 memory). See section 2.8.8.8.1.
6. Perform second pass of steps 4 through 7 at the target frequency, if applicable. See note 10 at the end of this section.
7. DRAM Data Training. See section 2.8.8.8 and all sub-sections:
   a. Receiver Enable Training. See sections 2.8.8.8.2.1 and 2.8.8.8.2.2.
   b. DQS Position Training.See section 2.8.8.8.2.3.
   c. MaxRdLat Training. See section 2.8.8.8.4.1.
8. Program Non-SPD timings to optimal values. See section 2.8.8.4 and all sub-sections.
9. The memory subsystem is ready for use.

Notes:

10. If DDR3 memory is used and the target frequency is greater than 400 MHz, then BIOS must perform two initialization passes for certain steps during the DCT/DRAM initialization; the first pass BIOS configures MEMCLK frequency to 400 Mhz, while the second pass BIOS configures MEMCLK frequency to the target frequency.

2.8.8.1 Phy and Controller Mode Configuration

To enable subsequent phy and controller register accesses to be routed correctly, Family 10h BIOS must do the following very early in POST:

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

2.8.8.2 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 unganged 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 section 2.8.8.5.

2.8.8.3 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 I/O hub, which obtains the information through a secondary device connected to the I/O hub through the SMBus. This secondary device communicates with the DIMM by means of the I^2C 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:

- Tcl: (CAS latency)
- Trc: Active-to-Active/Auto Refresh command period
- Trfc: Auto-Refresh-to-Active/Auto Refresh command period
- Trcd: Active-to-Read-or-Write delay
- Trrd: Active-Bank-A to-Active-Bank-B delay
- Tras: Active-to-Precharge delay
- Trp: Precharge time
- Tref: Refresh interval
- Trtp: Internal Read to Precharge command delay
- Twtr: Internal Write to Read command delay
- Twr: Write recovery time

Optimal cycle time is specified for each DIMM and is used to limit or determine bus frequency. See section 2.8.8.7 [DRAM Channel Frequency Change] on page 85 for more information on configuring the bus frequency.

2.8.8.4 Non-SPD ROM-Based Configuration

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

- DDR2 vs. DDR3 DRAM types.
- Mixed or non-mixed DIMMs (x4 with x8).
- Training delay values. See section 2.8.8.8 [DRAM Training] on page 86.
- 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.

2.8.8.4.1 Trdrd (Read to Read Timing)

The timing parameter Trdrd (see F2x[1, 0]8C[Trdrd]) account for bus turn-around time when a read is followed by a read to a different DIMM. The optimal values for Trdrd are platform and configuration specific and should be characterized for best performance. Prior to DRAM training, BIOS should program these parameters to the largest defined value. After DRAM training, BIOS should use the guidelines below to configure the rec-

ommended platform generic timing values after DDR training is complete:

- BIOS calculates Trdrd (in MEMCLKs) = CGDD / 2 + 3 clocks and programs F2x[1, 0]8C[Trdrd] with the converted field value. BIOS rounds fractional values down.
  - The Critical Gross Delay Difference (CGDD) for Trdrd on any given byte lane is the largest F2x[1, 0]9C_x[2B:10][DqsRcvEnGrossDelay] delay of any DIMM minus the F2x[1, 0]9C_x[2B:10][DqsRcvEnGrossDelay] delay of any other DIMM.

2.8.8.4.2 Twrwr (Write to Write Timing)

The timing parameter Twrwr (see F2x[1, 0]8C[Twrwr]]) account for bus turn-around time when a write is followed by a write to a different DIMM. The optimal values for Twrwr are platform and configuration specific and should be characterized for best performance. Prior to DRAM training, BIOS should program these parameters to the largest defined value; otherwise, BIOS should program Twrwr as follows:

- BIOS calculates Twrwr (in MEMCLKs) = CGDD / 2 + 3 clocks and programs F2x[1, 0]8C[Twrwr] with the converted field value. BIOS rounds fractional values down.
  - On any given byte lane, the largest F2x[1, 0]9C_x[302:301, 202:201, 102:101, 02:01]:F2x[1, 0]9C_x[303, 203, 103, 03][WrDatGrossDlyByte] delay of any DIMM minus the F2x[1, 0]9C_x[302:301, 202:201, 102:101, 02:01]:F2x[1, 0]9C_x[303, 203, 103, 03][WrDatGrossDlyByte] delay of any other DIMM is equal to the Critical Gross Delay Difference (CGDD) for Twrwr.

2.8.8.4.3 Twrrd (Write to Read DIMM Termination Turn-around)

The timing parameter Twrrd (see F2x[1, 0]8C[Twrrd]), account for bus turn-around time when a write is followed by a read to a different DIMM. The optimal values for Twrrd are platform and configuration specific and should be characterized for best performance. Prior to DRAM training, BIOS should program these parameters to the largest defined value; otherwise, BIOS should use the guidelines below to configure the recommended platform generic timing values after DDR training is complete:

- BIOS calculates Twrrd (in MEMCLKs) = CGDD / 2 - LD + 3 clocks and programs F2x[1, 0]8C[Twrrd] 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[Tcl] and F2x[1, 0]84[Tcwl]) which can be a negative or positive value.
  - For DDR2, the LD is always one clock.
  - On any given byte lane, the largest F2x[1, 0]9C_x[302:301, 202:201, 102:101, 02:01]:F2x[1, 0]9C_x[303, 203, 103, 03][WrDatGrossDlyByte] delay of any DIMM minus the F2x[1, 0]9C_x[2B:10][DqsRcvEnGrossDelay] delay of any other DIMM is equal to the Critical Gross Delay Difference (CGDD) for Twrrd.

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

The timing parameter TrwtTO (see F2x[1, 0]8C[TrwtTO]), ensures read-to-write data-bus turn-around time when a read is followed by a write to a different chip select. The optimal value for 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 values after DDR training is complete:

- BIOS calculates TrwtTO (in MEMCLKs) = CGDD / 2 + LD + 3 clocks and programs F2x[1, 0]8C[Twrrd] 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[Tcl] and F2x[1, 0]84[Tcwl]) which can be a negative or positive value.
- For DDR2, the LD is always one clock.

On any byte lane, the largest F2x[1, 0]9C_x[2B:10][DqsRcvEnGrossDelay] delay of any DIMM minus the F2x[1, 0]9C_x[302:301, 202:201, 102:101, 02:01]:F2x[1, 0]9C_x[303, 203, 103, 03][WrDatGrossDlyByte] delay of any other DIMM is equal to the Critical Gross Delay Difference (CGDD) for TrwtTO.

2.8.8.4.5 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 section [The TrwtTO (Read-to-Write Turnaround for Data, DQS Contention)] 2.8.8.4.4.

2.8.8.4.6 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 17 shows the DDR2 F2x[1, 0]94[tFAW] clock values used for various frequencies and page sizes.

Table 17: 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.

2.8.8.4.7 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 18 documents the ODT termination values for different DDR2 configurations. The DDR2 DIMM nominal termination resistance is controlled by F2x[1, 0]90[DramTerm].

Table 19 documents the ODT nominal (non-write) and dynamic termination resistance values for different DDR3 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 20 documents the ODT nominal (non-write) and dynamic termination resistance values for different DDR3 4 rank DIMM configurations. The BIOS enables nominal termination on even numbered ranks and disables nominal termination of odd numbered ranks of a 4 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 18: Processor and DDR2 DIMM ODT Settings

Rate	Number of DIMMs	Processor ODT	DIMM ODT
DDR2-400, DDR2-533, DDR2-667	1	75 ohms	75 ohms
DDR2-400, DDR2-533	2 or more	150 ohms	75 ohms
DDR2-667	2 or 3	150 ohms	75 ohms
DDR2-667	4	150 ohms	50 ohms
DDR2-800	1	75 ohms	75 ohms
DDR2-800	2 or more	150 ohms	50 ohms
DDR2-1066	1	75 ohms	75 ohms

Table 19: Processor and DDR3 DIMM ODT Settings

Number of DIMMs	Processor ODT	DIMM ODT (Rtt_Nom)	DIMM Dynamic ODT (Rtt_Wr)
1	60 ohms	60 ohms	Disabled
2 or 3	60 ohms	40 ohms	120 ohms
4	60 ohms	60 ohms	120 ohms

Table 20: Processor and QR-DDR3 DIMM ODT Settings

Number of DIMMs[1]	Processor ODT	DIMM ODT[2] (Rtt_Nom)	DIMM Dynamic ODT (Rtt_Wr)
1 or 2 QR	60 ohms	60 ohms	120 ohms
1 SR or DR	60 ohms	30 ohms	120 ohms

1. QR = Quad Rank. SR, DR = Single rank, dual rank.
2. BIOS enables Rtt_Nom only on ranks S0 and S2 of a 4 rank DIMM.

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 rank1 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.

2.8.8.4.8 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. Table 21, Table 22, Table 23, and Table 24 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 section 2.8 [DRAM Controllers (DCTs)] on page 64 for an overview of the DIMM and memory bus speed support.

Table 21: DDR2 Unbuffered DIMM Address Timings and Drive Strengths for the AM2r2 Package

DDR Type-Rate	DIMM0[1]	DIMM1[1]	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00[2]
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
DDR2-1066	-	DRx8	2T	0020_2520h	X011_3222h
		SRx8			
		SRx16			

1. SR = Single Rank DIMM
   DR = Dual Rank DIMM
   any = SR, DR
2. See Table 18 on page 74 for DDR2 ProcODT settings.

Table 22: DDR2 Registered DIMM Address Timings and Drive Strengths for Fr2(1207) Package (4 DIMMs per channel)

DDR Type-Rate	DIMM0[1]	DIMM1[1]	DIMM2[1]	DIMM3[1]	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00[2]
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 or DRx8	SR or DRx8	1T	0037_0000h	X011_1222h
DDR2-667	-	-	any	DRx4	1T	0037_2F00h	X011_1222h
DDR2-667	-	-	DRx4	any	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	-	-	DRx4	any	1T	0037_2F00h	X011_1222h
DDR2-800	-	-	any	DRx4	1T	0037_2F00h	X011_1222h

1. Any = SR or DR.
2. See Table 18 on page 74 for DDR2 ProcODT settings.

Table 23: DDR2 Registered DIMM Address Timings and Drive Strengths for the Fr2(1207) Package (2 DIMMs per channel)

DDR Type-Rate	DIMM0[1]	DIMM1[1]	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00[2]
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

1. SR = Single Rank DIMM
   DR = Dual Rank DIMM
   QR = Quad Rank DIMM
   any = SR, DR, or QR
2. See Table 18 on page 74 for DDR2 ProcODT settings.

Table 23: DDR2 Registered DIMM Address Timings and Drive Strengths for the Fr2(1207) Package (2 DIMMs per channel)

DDR Type-Rate	DIMM0[1]	DIMM1[1]	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00[2]
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	any	DRx4	1T	0037_2F00h	X011_1222h
DDR2-667	DRx4	any	1T	0037_2F00h	X011_1222h
DDR2-667	QR	QR	1T	002F_2F00h	X033_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. SR = Single Rank DIMM
   DR = Dual Rank DIMM
   QR = Quad Rank DIMM
   any = SR, DR, or QR
2. See Table 18 on page 74 for DDR2 ProcODT settings.

Table 24: DDR3 Unbuffered DIMM Address Timings and Drive Strengths

DDR Type-Rate	DIMM0	DIMM1	Timing Mode	F2x[1, 0]9C_x04	F2x[1, 0]9C_x00
DDR3-800	-	SR-x16	1T	0037_3737h	2022_3222h
DDR3-800	-	SR-x8	1T	0037_3737h	2022_3222h
DDR3-800	-	DR-x8	1T	0037_3737h	2022_3222h
DDR3-800	SR-x16	SR-x16	1T	0037_3737h	2022_3322h
DDR3-800	SR-x8	SR-x8	1T	0037_3737h	2022_3322h
DDR3-800	DR-x8	DR-x8	1T	0037_3737h	2022_3322h
DDR3-800	SR-x16	SR-x8	1T	0037_3737h	2022_3322h
DDR3-800	SR-x8	SR-x16	1T	0037_3737h	2022_3322h
DDR3-800	SR-x16	DR-x8	1T	0037_3737h	2022_3322h
DDR3-800	DR-x8	SR-x16	1T	0037_3737h	2022_3322h
DDR3-800	SR-x8	DR-x8	1T	0037_3737h	2022_3322h
DDR3-800	DR-x8	SR-x8	1T	0037_3737h	2022_3322h
DDR3-1066	-	SR-x16	1T	0037_3737h	2022_3222h
DDR3-1066	-	SR-x8	1T	0037_3737h	2022_3222h
DDR3-1066	-	DR-x8	1T	0037_3737h	2022_3222h

Table 24: DDR3 Unbuffered DIMM Address Timings and Drive Strengths

DDR3-1066	SR-x16	SR-x16	1T	0035_3737h	2022_3322h
DDR3-1066	SR-x8	SR-x8	1T	0035_3737h	2022_3322h
DDR3-1066	DR-x8	DR-x8	1T	0035_3737h	2022_3322h
DDR3-1066	SR-x16	SR-x8	1T	0035_3737h	2022_3322h
DDR3-1066	SR-x8	SR-x16	1T	0035_3737h	2022_3322h
DDR3-1066	SR-x16	DR-x8	1T	0035_3737h	2022_3322h
DDR3-1066	DR-x8	SR-x16	1T	0035_3737h	2022_3322h
DDR3-1066	SR-x8	DR-x8	1T	0035_3737h	2022_3322h
DDR3-1066	DR-x8	SR-x8	1T	0035_3737h	2022_3322h
DDR3-1333	-	SR-x16	1T	0037_3737h	2022_3222h
DDR3-1333	-	SR-x8	1T	0037_3737h	2022_3222h
DDR3-1333	-	DR-x8	1T	0037_3737h	2022_3222h
DDR3-1333	SR-x16	SR-x16	2T	0000_3737h	2022_3322h
DDR3-1333	SR-x8	SR-x8	2T	0000_3737h	2022_3322h
DDR3-1333	DR-x8	DR-x8	2T	0000_3737h	2022_3322h
DDR3-1333	SR-x16	SR-x8	2T	0000_3737h	2022_3322h
DDR3-1333	SR-x8	SR-x16	2T	0000_3737h	2022_3322h
DDR3-1333	SR-x16	DR-x8	2T	0000_3737h	2022_3322h
DDR3-1333	DR-x8	SR-x16	2T	0000_3737h	2022_3322h
DDR3-1333	SR-x8	DR-x8	2T	0000_3737h	2022_3322h
DDR3-1333	DR-x8	SR-x8	2T	0000_3737h	2022_3322h

2.8.8.5 DRAM Device Initialization

After the DCT registers are programmed (see sections 2.8.8.3 and 2.8.8.4), BIOS initializes the DRAM using either a hardware or BIOS controlled sequence. See sections 2.8.8.5.1 and 2.8.8.5.2 for more information on BIOS controlled initialization. To initiate the hardware sequence, BIOS does the following:

1. Program F2x[1, 0]90[InitDram] = 1. See note.

Note: BIOS must observe additional requirements for changing the PLL frequency by setting F2x[1, 0]90[InitDram]. See section 2.8.8.7 [DRAM Channel Frequency Change] on page 85 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]88[Twr].
- 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[PchgPDModeSel].
- 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.

2.8.8.5.1 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 10 ns.
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 section 2.8.8.7 [DRAM Channel Frequency Change] on page 85 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 Trfc.

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]88[Twr].
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] = 1.
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.

2.8.8.5.2 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 section 2.8.8.7 [DRAM Channel Frequency Change] on page 85 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 2.8.8.5.2.1 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[PchgPDModeSel].
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.

2.8.8.5.2.1 Software Control Word Initialization

DDR3 data register and clock driver 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 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.

Send RCW(n) command for DDR3 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 *target chip selects*.
3. Set SendControlWord = 1.

4. Wait for SendControlWord = 0.

Based on the DIMM raw card version, BIOS programs RC0 to RC5 for the DIMM according to the information in the following table:

Table 25. DDR3 RDIMM Register Control Word Values

Control Word	Raw Card Version[1]					
	A (1Rx8)	**B (2Rx8)**	**C (1Rx4)**	**D (2Rx4)**	**E (2Rx4)**	**F (4Rx4)**
RC0	0h	0h	0h	0h	0h	0h
RC1	Ch	0h	Ch	0h	0h	0h
RC2[2]	0X00b	0X00b	0X00b	0X00b	0X00b	0X00b
RC3	0h	5h	5h	Ah	Ah	Ah
RC4	0h	0h	5h	5h	5h	5h
RC5	0h	0h	5h	5h	5h	5h
RC8	0h	0h	0h	0h	0h	0h
RC9	0Ch	0Ch	0Ch	0Ch	0Ch	0Ch

Notes:
1. JEDEC raw card version is determined by reading SPD byte 62.
   R=rank.
2. X=0b for 1 or 2 logical RDIMMs per channel. (buffer term is 100 ohms)
   X=1b for 3 or 4 logical RDIMMs per channel.(buffer term is 150 ohms)

2.8.8.6 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 section 2.8.8.7) and before any memory training, BIOS must perform phy fence training for each channel using the following steps:

1. BIOS first programs a seed value to the phase recovery engine registers.
2. Set F2x[1, 0]9C_x08[PhyFenceTrEn]=1.
3. Wait 200 MEMCLKs.
4. Clear F2x[1, 0]9C_x08[PhyFenceTrEn]=0.
5. BIOS reads the phase recovery engine registers F2x[1, 0]9C_x[51:50] and F2x[1, 0]9C_x52.
6. Calculate the average value and subtract 8.
7. Write the value to F2x[1, 0]9C_x0C[PhyFence].
8. BIOS rewrites F2x[1, 0]9C_x04, DRAM Address/Command Timing Control Register delays for both channels.

2.8.8.7 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.
- 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 unganged 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.

2.8.8.8 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 2.8.8.5 [DRAM Device Initialization] on page 79). 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 section 2.8 [DRAM Controllers (DCTs)] on page 64) then the training algorithms are done in ganged mode. Likewise, if the DCTs are unganged then the training is done unganged. 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.

DRAM training of the ECC byte lanes is accomplished after the data lanes are trained. This is described in section 2.8.8.8.3 [ECC Byte Lane Training] on page 97.

2.8.8.8.1 DDR3 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 section 2.8.8.8.2 [DDR2 DRAM Training] on page 91. 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. A two pass levelization is necessary if the target MEMCLK frequency is different than 400MHz.

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.

There are two methods to accomplish write levelization training: phy assisted or BIOS based training.

2.8.8.8.1.1 Phy Assisted Write Levelization

Phy assisted write levelization involves using the DDR phy to detect where the edge of DQS is with respect to MEMCLK on the DIMM for write accesses. The following describes the steps used in phy assisted write levelization training:

Pass 1:
- Program F2x[1, 0]9C_x08[WrtLlTrEn]=0.
- Configure the DRAM interface for a 400MHz MEMCLK (DDR3-800) frequency.
- Disable auto refresh by configuring F2x[1, 0]8C[DisAutoRefresh] = 1.
- Disable ZQ calibration short command by configuring F2x[1, 0]94[ZqcsInterval] = 00b.

For each DIMM or chip select pair of a quad rank DIMM:
1. Specify the target DIMM that is to be trained by programming F2x[1, 0]9C_x08[TrDimmSel].
2. Prepare the DIMMs for write levelization using DDR3-defined MR commands.
   - 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.
   - Configure the DCT to send initialization MR commands to the target DIMM by programming the F2x[1, 0]7C register using the following steps.
     - Program F2x[1, 0]7C[MrsChipSel[2:0]] for the current rank to be trained.
     - Program F2x[1, 0]7C[MrsBank[2:0]] for the appropriate internal DRAM register that defines the required DDR3-defined function for write levelization.
     - Program F2x[1, 0]7C[MrsAddress[15:0]] to the required DDR3-defined function for write levelization.
     - Program F2x[1, 0]7C[SendMrsCmd]=1 to initiate the command to the specified DIMM.
     - Wait for F2x[1, 0]7C[SendMrsCmd] to be cleared by hardware.
   - Using the above DIMM initialization steps, configure the target DIMMs normally as follows:
     - All ranks of the target DIMM are set to write levelization mode.
     - Enable the output driver of the first rank of the target DIMM.
     - Disable the output drivers of all other ranks for the target DIMM.
     - Write leveling mode enable and output driver enable/disable can use a single MRS command; one to each rank separately.
     - For two DIMMs in the system, program the Rtt_Nom value for the target DIMM to the weakest termination (RZQ/2) using a send MR command. Otherwise, configure the target DIMM normally. See section 2.8.8.4.7 for more information on normal ODT settings.
   - Configure the non-target DIMM normally (See section 2.8.8.4.7).
3. After the DIMMs are configured, BIOS waits 40 MEMCLKs to satisfy DDR3-defined internal DRAM timing.
4. Configure the processor's DDR phy for write levelization training:
   - Program F2x[1, 0]9C_x08[WrLvOdt[3:0]] to the proper ODT settings for the current memory subsystem configuration.
   - Program F2x[1, 0]9C_x08[WrLvOdtEn]=1.
   - Wait 10 MEMCLKs to allow for ODT signal settling.

- Program an initialization value to registers F2x[1, 0]9C_x[51:50] and F2x[1, 0]9C_x52 to set the gross and fine delay for all the byte lane fields. If the target frequency is different than 400MHz, BIOS must execute two training passes for each DIMM.
  - For Pass 1 at a 400MHz MEMCLK frequency, use an initial total delay value specified in 2.8.8.8.1.2.2 [Write Leveling Seed Value] on page 91.
  - For Pass 2, at the target MEMCLK frequency, use the results of the Pass 1 as the initial delay values for the seed. Note: if BIOS intends to execute phy assisted DQS receiver enable training (2.8.8.8.2.1 [Phy Assisted DQS Receiver Enable Training] on page 91) then Pass 2 of this section must not be executed until after Pass 1 of 2.8.8.8.2.1 [Phy Assisted DQS Receiver Enable Training] on page 91 is complete.
- Program F2x[1, 0]9C_x08[WrtLvTrMode]=0 for phy assisted training.
- Program F2x[1, 0]9C_x08[TrNibbleSel]=0[1].

5. Begin write levelization training:
   - Program F2x[1, 0]9C_x08[WrtLlTrEn]=1.
   - Wait 200 MEMCLKs. If executing Pass 2, wait 32 MEMCLKs.
   - Program F2x[1, 0]9C_x08[WrtLlTrEn]=0.
   - Read from registers F2x[1, 0]9C_x[51:50] and F2x[1, 0]9C_x52 to get the gross and fine delay settings for the target DIMM and save these values.
6. Configure DRAM Phy Control Register so that the phy stops driving write levelization ODT.
   - Wait 10 MEMCLKs to allow for ODT signal settling.
7. Program the target DIMM back to normal operation by configuring the following: (see step #2 above)
   - Configure all ranks of the target DIMM for normal operation.
   - Enable the output drivers of all ranks of the target DIMM.
   - For a two DIMM system, program the Rtt value for the target DIMM to the normal operating termination: F2x[1, 0]84[Dram Term]= 011b (RZQ/6 = 40 ohms).

If the target MEMCLK frequency is different than 400MHz, execute Pass 2, steps #8 through #11 below; else execute step #12:

Pass 2:

8. Prepare the memory subsystem for the target MEMCLK frequency. Note: BIOS must program both DCTs to the same frequency.
   - Program F2x[1, 0]90[EnterSelfRefresh]=1.
   - Wait until the hardware resets F2x[1, 0]90[EnterSelfRefresh]=0.
   - Program F2x[1, 0]9C_x08[DisAutoComp] =1.
   - Program F2x[1, 0]94[MemClkFreqVal] = 0.
   - Program F2x[1, 0]94[MemClkFreq] to specify the target MEMCLK frequency.
   - Program F2x[1, 0]94[MemClkFreqVal] = 1.
   - Wait until F2x[1, 0]94[FreqChgInProg]=0.
   - Program F2x[1, 0]9C_x08[DisAutoComp] =0.
   - Program F2x[1, 0]90[ExitSelfRef]=1 for both DCTs.
   - Wait until the hardware resets F2x[1, 0]90[ExitSelfRef]=0.
   - Perform Phy Fence retrainingSee section 2.8.8.6.
9. Configure the DCT to send initialization MR commands: BIOS must reprogram Twr, Tcwl, and Tcl based on the new MEMCLK frequency. Program F2x[1, 0]7C similar to step #2 in Pass 1 above for the new DIMM values.
10. Multiply the previously saved delay values in Pass 1, step #5 by (target frequency)/400 to find the gross and fine delay initialization values at the target frequency. Use these values as the initial seed values when executing Pass 2, step #4.
11. For each DIMM, execute steps #1 through #7 in Pass 1 above.
12. For each DIMM, program the gross and fine delays for each field of registers F2x[1, 0]9C_x[45:30] with

the corresponding final saved delay settings.
13. Program F2x[1, 0]8C[DisAutoRefresh] = 0.
14. Program F2x[1, 0]94[ZqcsInterval] to the proper interval for the current memory configuration.

Note 1: If BIOS needs to train based on both the lower and upper nibbles it can program the F2x[1, 0]9C_x08[TrNibbleSel] register to specify which nibbles (0=lower, 1=upper) the training uses and then repeat the Pass 1 steps above. BIOS then averages the training values for the nibbles of each byte and uses the average for the delay settings.

2.8.8.8.1.2 BIOS Based Write Levelization Training

BIOS controlled write levelization involves using the BIOS to detect where the edge of DQS is with respect to the memory clock on the DIMM for write accesses to each lane. In the steps below, a lane is used to describe either a 4-bit wide data group or an 8-bit wide data group, each with its own write DQS timing control. Normally, an 8-bit lane size is used.

The following describes the steps used for BIOS controlled write levelization training:

Pass 1:
- Program F2x[1, 0]9C_x08[WrtLlTrEn]=0.
- Configure the DRAM interface for a 400MHz MEMCLK (DDR3-800).
- Disable auto refresh by configuring F2x[1, 0]8C[DisAutoRefresh] = 1.
- Disable ZQ calibration short command by configuring F2x[1, 0]94[ZqcsInterval] = 00b.

For each chip select pair :
1. Specify the target DIMM that is to be trained by programming F2x[1, 0]9C_x08[TrDimmSel].
2. Prepare the DIMMs for write levelization using DDR3-defined MR commands. Execute Pass 1, steps #2 and #3 in section 2.8.8.8.1.1 [Phy Assisted Write Levelization] on page 87.
3. Configure the phy for write levelization training:
   - Program F2x[1, 0]9C_x08[WrLvOdt[3:0]] to the proper ODT settings for the current memory subsystem configuration.
   - Program F2x[1, 0]9C_x08[WrLvOdtEn]=1.
   - Wait 10 MEMCLKs to allow for ODT signal settling.
   - Program an initialization value to registers F2x[1, 0]9C_x[51:50] and F2x[1, 0]9C_x52 to set the gross and fine delay for all the byte lane fields. If the target frequency is different than 400MHz, BIOS must execute two training passes for each DIMM.
     - For Pass 1 at a 400MHz MEMCLK frequency, use an initial total delay value specified in 2.8.8.8.1.2.2 [Write Leveling Seed Value] on page 91.
     - For Pass 2, at the target MEMCLK frequency, use the results of the Pass 1 as the initial delay values for the seed.
   - Program F2x[1, 0]9C_x08[WrtLvTrMode]=1 for BIOS controlled training.
   - Program F2x[1, 0]9C_x08[TrNibbleSel]=0[1].
4. Perform write leveling of the devices on the DIMM as described in section 2.8.8.8.1.2.1.
5. Program F2x[1, 0]9C_x08[WrLvOdtEn]=0 so that the phy stops driving write levelization ODT.
   - Wait 10 MEMCLKs to allow for ODT signal settling.
6. Program the target DIMM back to normal operation by configuring the following (See section 2.8.8.8.1.1 [Phy Assisted Write Levelization] on page 87 Pass 1, step #2):
   - Configure all ranks of the target DIMM for normal operation.
   - Enable the output drivers of all ranks of the target DIMM.
   - For a two DIMM system, program the Rtt value for the target DIMM to the normal operating termination: F2x[1, 0]84[Dram Term]= 011b (RZQ/6 = 40 ohms).

If the target MEMCLK frequency is different than 400MHz, execute Pass 2, steps #7 through #10 below; else execute step #11:

Pass 2:

7. Prepare the memory subsystem for the target MEMCLK frequency. Note: BIOS must change the frequency on both channels together. Both channels must be at the same frequency.
   - Program F2x[1, 0]90[EnterSelfRefresh]=1.
   - Wait until the hardware resets F2x[1, 0]90[EnterSelfRefresh]=0.
   - Program F2x[1, 0]9C_x08[DisAutoComp] =1.
   - Program F2x[1, 0]94[MemClkFreqVal] = 0.
   - Program F2x[1, 0]94[MemClkFreq] to specify the target MEMCLK frequency.
   - Program F2x[1, 0]94[MemClkFreqVal] = 1.
   - Wait until F2x[1, 0]94[FreqChgInProg]=0.
   - Program F2x[1, 0]9C_x08[DisAutoComp] =0.
   - Program F2x[1, 0]90[ExitSelfRef]=1 for both DCTs.
   - Wait until the hardware resets F2x[1, 0]90[ExitSelfRef]=0.
   - Perform Phy Fence retraining See section 2.8.8.6.
8. Configure the DCT to send initialization MR commands: BIOS must reprogram Twr, Tcwl, and Tcl based on the new MEMCLK frequency. Program F2x[1, 0]7C similar to step #2, Pass 1 in section 2.8.8.8.1.1 [Phy Assisted Write Levelization] on page 87 for the new DIMM values.
9. Multiply the previously saved delay values by (target frequency)/400 MHz to find the coarse delay and fine delay seed at the target frequency which is used in the second pass.
10. For each DIMM, execute steps #1 through #6 in Pass 1 above.
11. each DIMM, program the gross and fine delay fields for each byte lane of registers F2x[1, 0]9C_x[45:30] with the corresponding saved final delay settings.
12. Program F2x[1, 0]8C[DisAutoRefresh] = 0.
13. Program F2x[1, 0]94[ZqcsInterval] to the proper interval for the current memory configuration.

Note: If BIOS needs to train based on both the lower and upper nibbles it can program the F2x[1, 0]9C_x08[TrNibbleSel] register to specify which nibbles (0=lower, 1=upper) the training uses and then repeat the Pass 1 steps above. BIOS then averages the training values for the nibbles of each byte and uses the average for the delay settings.

2.8.8.8.1.2.1 Write Leveling

BIOS performs write leveling of the devices on a DIMM using the following steps, for different total delay values:

1. Program F2x[1, 0]9C_x08[WrtLlTrEn]=1.
2. Wait 32 MEMCLKs.
3. Program F2x[1, 0]9C_x08[WrtLlTrEn]=0.
4. BIOS reads from F2x[1, 0]9C_x53[WrtLvErr[8:0]] to get the current state of each byte lane field.
5. Compare the state of each bit of F2x[1, 0]9C_x53[WrtLvErr[8:0]]; if the bit=0, increment the total delay field for the corresponding byte lane of registers F2x[1, 0]9C_x[51:50] and F2x[1, 0]9C_x52; if the bit=1, decrement the total delay field for the corresponding byte lane in registers F2x[1, 0]9C_x[51:50] and F2x[1, 0]9C_x52. If the error bit is steady state 0, increment the gross delay value until a transition is seen. At this point, begin decrementing the fine delay until another transition is seen. Likewise, if the error bit is steady state 1, decrement the gross delay value until a transition is seen, then, begin incrementing the fine delay until another transition is seen.
6. BIOS loops back to step #1 with the adjusted gross and fine delay settings until it sees a 0 to 1 or 1 to 0

transition.
7. Save the total delay values for registers F2x[1, 0]9C_x[51:50] and F2x[1, 0]9C_x52.

2.8.8.8.1.2.2 Write Leveling Seed Value

BIOS uses a seed value for the first pass (400MHz) of write leveling using the following guidelines:

- For Unbuffered DIMMs, the seed for the total delay is 1Fh. This represents a 1UI (UI=.5MEMCLK) delay.
- For Registered DIMMs, if bit 0 of RCW2 (programmed with A3) is clear, then the seed for the total delay is 41h.
- For Registered DIMMs, if bit 0 of RCW2 is set, then the seed for the total delay is 51h.

Note: The seed value represents the actual clock delay and is platform dependent. The platform vendor may need to characterize and adjust this value for proper write levelization training. The seed delay value must fall within +/- 1.20 ns, including PVT and jitter, of the measured clock delay.

2.8.8.8.2 DDR2 DRAM Training

DDR2 DRAM training is used to determine when to enable the processor's DQS receivers for reads and to position the DQS in the center of the data eye for both reads and writes. DDR2 training is performed at the desired memory clock rate.

Part of the DDR2 training process requires BIOS to calculate a latency component called F2x[1, 0]78[MaxRdLatency]. This is described in section 2.8.8.8.4 [Calculating MaxRdLatency] on page 98.

2.8.8.8.2.1 Phy Assisted DQS Receiver Enable Training

The following section describes the phy assisted DQS receiver enable training algorithm. Phy assisted DQS receiver enable training involves using the DDR phy to determine the values required to program the DRAM DQS Receiver Enable Timing Control registers (see F2x[1, 0]9C_x[2B:10]). This procedure optimizes the timing used to turn on the receiver enables when doing reads from the DRAM. Receiver enable training is performed after write levelization (see section 2.8.8.8.1 [DDR3 Training] on page 86).

Phy assisted training is accomplished on a per channel, per DIMM basis. It is not necessary to train each rank of a specific DIMM as the skew between ranks on the same DIMM is negligible. For DDR2, a single training pass is required for each DIMM. For DDR3, a two pass DQS receiver enable training is necessary if the target MEMCLK frequency is different than 400MHz.

The following describes the steps used for the phy assisted receiver enable training:

Pass 1:

- For DDR3, configure the DRAM interface for a 400MHz MEMCLK (DDR3-800). For DDR2, configure the DRAM interface for the target frequency.
- Disable auto refresh by configuring F2x[1, 0]8C[DisAutoRefresh] = 1.
- Disable ZQ calibration short command by configuring F2x[1, 0]94[ZqcsInterval] = 00b.

For each DIMM:
1. Prepare the DIMMs for training:
   - Specify the DIMM to be trained by configuring [The DRAM Phy Control Register] F2x[1, 0]9C_x08[TrDimmSel].

- For DDR3, configure the first chip select of the target DIMM for 8-beat burst length mode using DDR3-defined MR commands.
  - Configure the DCT to send initialization MR commands to the target DIMM by programming the F2x[1, 0]7C register using the following steps.
    - Program F2x[1, 0]7C[MrsChipSel[2:0]] for the current rank to be trained.
    - Program F2x[1, 0]7C[MrsBank[2:0]] for the appropriate internal DRAM register that defines the required DDR3-defined function for training.
    - Program F2x[1, 0]7C[MrsAddress[15:0]] to the required DDR3-defined function for training.
    - Program F2x[1, 0]7C[SendMrsCmd]=1 to initiate the command to the specified DIMM.
    - Wait for F2x[1, 0]7C[SendMrsCmd] to be cleared by hardware.
  - Wait 20 MEMCLKs to allow for command completion.
- For DDR2, no additional DIMM configuration is needed for training.

2. Prepare the phy for DQS receiver enable training.
   - Program the DRAM Phase Recovery Control Registers (see F2x[1, 0]9C_x[51:50] and F2x[1, 0]9C_x52) with a gross and fine delay seed value. The following steps are taken to determine the seed values needed to program the DRAM Phase Recovery Control Registers for each byte lane:
     - For Pass 1:
       - BIOS starts with the delay value obtained from the first pass of write levelization training that was done in section 2.8.8.8.1 [DDR3 Training] on page 86 and add a delay value of 26h. This corresponds to a 1.2UI (UI=.5MEMCLK) delay and is called SeedTotal.
       - Next, determine the gross component of SeedTotal. SeedGrossPass1=SeedTotal DIV 32.
       - Then, determine the fine delay component of SeedTotal. SeedFinePass1=SeedTotal MOD 32.
       - Use SeedGrossPass1 to determine SeedPreGrossPass1: SeedPreGrossPass1=1 if SeedGrossPass1 is odd; SeedPreGrossPass1=2 if SeedGrossPass1 is even. This allows the phase recovery engine a positive and negative adjustment range.
       - The last term that BIOS needs to calculate for the seed is called SeedPass1Remainder. SeedPass1Remainder=SeedGrossPass1-SeedPreGrossPass1. BIOS saves this value which is used to determine the final delay value. This is necessary because the phy does not have the full range of delay adjustment.
       - BIOS programs registers F2x[1, 0]9C_x[51:50] and F2x[1, 0]9C_x52) with SeedPreGrossPass1 and SeedFinePass1 from the preceding steps.
     - For Pass2, for each byte lane, BIOS uses executes the following steps and uses the results obtained from Pass 1 to determine the seed value. These steps are needed only if the target MEMCLK frequency is not 400MHz.
       - Determine the total seed based on the delay value used in Pass1: SeedTotalPass2=(Pass1DqsRcvEnDly*target frequency)/400.
       - Next, determine the gross component of SeedTotalPass2: SeedGrossPass2=SeedTotalPass2 DIV 32.
       - Then, BIOS determines the fine delay component of SeedTotalPass2: SeedFinePass2=SeedTotalPass2 MOD 32.
       - Use SeedGrossPass2 to determine SeedPreGrossPass2: SeedPreGrossPass2=1 if SeedGrossPass2 is odd; SeedPreGrossPass2=2 if SeedGrossPass2 is even. This allows the phase recovery engine a positive and negative adjustment range.
       - BIOS calculates SeedPass2Remainder. SeedPass2Remainder=SeedGrossPass2-SeedPreGrossPass2. BIOS saves this value which is used to determine the final delay value. This is necessary because the phy does not have the full range of delay adjustment.
       - BIOS programs registers F2x[1, 0]9C_x[51:50] and F2x[1, 0]9C_x52) with SeedPreGrossPass2 and SeedFinePass2 from the preceding steps.
     - For DDR2, BIOS only executes 1 pass for each DIMM. BIOS uses a seed value of 26h. This corresponds to a 1.2UI (UI=.5MEMCLK) delay.
3. BIOS initiates the phy assisted receiver enable training by programming F2x[1, 0]9C_x08[Dqs-

RcvTrEn]=1.

4. BIOS begins sending out of back-to-back reads to create a continuous stream of DQS edges on the DDR interface.
   - Program F2x11C[MctPrefReqLimit] = 01Fh.
   - BIOS sends 32 read requests to the target rank. The reads must be to consecutive cache lines (i.e. 64 bytes apart) and must not cross a naturally aligned 4 Kbyte boundary. To generate the needed continuous read or write data streams for phy assisted receiver enable training, see section 2.8.8.8.5 [Continuous Pattern Generation] on page 100.
     Note: the phy implements a counter so that phase recovery engine is halted during down time between streams of DQS edges.
5. Repeat step #4 two more times.
6. Wait 200 MEMCLKs.
7. Program [The DRAM Phy Control Register] F2x[1, 0]9C_x08[DqsRcvTrEn]=0 to stop the DQS receive enable training.
8. Read from DRAM Phase Recovery Control Registers F2x[1, 0]9C_x[51:50] and F2x[1, 0]9C_x52 to get the gross and fine delay values.
9. Calculate the corresponding final delay values: For Pass 1, F2x[1, 0]9C_x[2B:10][DqsRcvEnGrossDelay] = SeedPass1Remainder+F2x[1, 0]9C_x[52:50]PhRecGrossDlyByte; F2x[1, 0]9C_x[2B:10][DqsRcvEnFineDelay]=F2x[1, 0]9C_x[52:50]PhRecFineDlyByte. For Pass 2, F2x[1, 0]9C_x[2B:10][DqsRcvEnGrossDelay] = SeedPass2Remainder+F2x[1, 0]9C_x[52:50]PhRecGrossDlyByte; F2x[1, 0]9C_x[2B:10][DqsRcvEnFineDelay]=F2x[1, 0]9C_x[52:50]PhRecFineDlyByte.
10. Save the delay values. Note: if BIOS is required to do nibble based write levelization and DQS receive enable training (for example, on x4 DIMM systems), repeat steps #2 through #9 above for the other nibbles by configuring [The DRAM Phy Control Register] F2x[1, 0]9C_x08[TrNibbleSel].

Perform steps #11 through #17 below if the system is using DDR3 type DIMMs at a frequency other than 400MHz else execute steps #14 through #17 only.

Pass 2:

11. Prepare the memory subsystem for the target MEMCLK frequency. Note: Both channels must be at the same frequency.
    - Program F2x[1, 0]90[EnterSelfRefresh]=1.
    - Wait until the hardware resets F2x[1, 0]90[EnterSelfRefresh]=0.
    - Program F2x[1, 0]94[MemClkFreqVal] = 0.
    - Program F2x[1, 0]94[MemClkFreq] to specify the target MEMCLK frequency.
    - Program F2x[1, 0]94[MemClkFreqVal] = 1.
    - Wait until F2x[1, 0]94[FreqChgInProg]=0.
    - Program F2x[1, 0]90[ExitSelfRef]=1 for both DCTs.
    - Wait until the hardware resets F2x[1, 0]90[ExitSelfRef]=0.
12. Configure the DCT to send initialization MR commands: BIOS must reprogram Twr, Tcwl, and Tcl based on the new MEMCLK frequency. Program F2x[1, 0]7C similar to step #2, Pass 1 in section 2.8.8.8.1.1 [Phy Assisted Write Levelization] on page 87 for the new DIMM values.
13. For each DIMM, execute steps #2 through #9 in Pass 1 above.
14. For DDR3, configure the target DIMM back to normal operation using MRS commands.
    - Wait 20 MEMCLKs to allow for command completion.
15. Program [The DRAM DQS Receiver Enable Timing Control Registers] F2x[1, 0]9C_x[2B:10] with the corresponding final delay values: F2x[1, 0]9C_x[2B:10][DqsRcvEnGrossDelay] = SeedPass2Remainder+F2x[1, 0]9C_x[52:50]PhRecGrossDlyByte; F2x[1, 0]9C_x[2B:10][DqsRcvEnFineDelay]=F2x[1, 0]9C_x[52:50]PhRecFineDlyByte.
16. Program F2x[1, 0]8C[DisAutoRefresh] = 0.

17. Program F2x[1, 0]94[ZqcsInterval] to the proper interval for the current memory configuration.

2.8.8.8.2.2 BIOS Based DQS Receiver Enable Training

This section describes the BIOS algorithm used to determine the values required to program the DRAM DQS Receiver Enable Timing Control registers (see F2x[1, 0]9C_x[2B:10]) for read DQS receiver enable position training. DQS receiver enable training determines when to enable the DQS receivers for reads on each DIMM and is accomplished in two training phases. The first phase is used to detect when the DIMM starts driving the DDR-defined read preamble phase of a read transaction. The second phase is used to determine the effective width of the read preamble and is optional for DDR2. The second training phase for DQS receiver enable is used to optimally place the receiver enable setting in the center of the read preamble. It can be used to obtain better performance margin in some circumstances. The second phase must be done after DQS position training is complete. See 2.8.8.8.2.3 [DQS Position Training] on page 96 for details on how to train the DRAM for proper DQS position.

DQS receiver enable training is done per channel, per rank, per lane. The lane size is 8-bits. When the processor revision supports less control resolution than the resolution of the training, then BIOS performs a reduction operation when processing the training results, e.g. averaging results of training two chip selects which share a per-DIMM timing control.

The first phase of DQS receiver training is performed using the following steps:

For each channel:

1. Program F2x[1, 0]9C_x[302:301, 202:201, 102:101, 02:01]:F2x[1, 0]9C_x[303, 203, 103, 03] to 00h for all lanes.
2. Program F2x[1, 0]9C_x[306:305, 206:205, 106:105, 06:05] and F2x[1, 0]9C_x[307, 207, 107, 07] to 2Fh for all lanes.
3. Program F2x[1, 0]78[DqsRcvEnTrain]=1.
4. Select two test addresses for each rank present. The addresses must be cache line (64 byte) aligned and separated by 2Meg starting with the first rank.
5. Write one cache line where each byte is 55h to the first test address for each rank.
6. Write one cache line where each byte is AAh to the second test address for each rank.
7. For each rank:
8. For all lanes, 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. For the start value and each subsequent total delay value in F2x[1, 0]9C_x[2B:10] do the following:
    a. Program F2x[1, 0]78[MaxRdLatency] with the current greatest value of F2x[1, 0]9C_x[2B:10]. See section 2.8.8.8.4 [Calculating MaxRdLatency] on page 98.
    b. 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.
    c. Reset the read pointer in the DRAM controller receive FIFO by writing the current corresponding DQS receiver enable delay settings to each lane in each corresponding F2x[1, 0]9C_x[2B:10] register.
    d. 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.
    e. Save each DQS receiver enable settings that passes for both read patterns. Continue to step 8 below.
    f. Increment the current total delay DQS receiver enable setting by one for each failing lane in F2x[1, 0]9C_x[2B:10] and repeat steps a through g. The total delay is the sum of the gross and fine delay

fields.

9. For each DIMM (chip select pair):
   - Program each F2x[1, 0]9C_x[2B:10] register with the first F2x[1, 0]9C_x[2B:10] register settings that passed for all ranks in the steps above plus 0.5 MEMCLK.
   - Save the first DQS receiver enable delay settings that pass for all ranks. This is used in phase two below.
10. Program F2x[1, 0]78[MaxRdLatency] with the current greatest value of F2x[1, 0]9C_x[2B:10]. See section 2.8.8.8.4 [Calculating MaxRdLatency] on page 98.
11. Program F2x[1, 0]78[DqsRcvEnTrain]=0.

Before completing the DQS receiver enable training, BIOS must complete the DQS position training described in the next section 2.8.8.8.2.3 [DQS Position Training] on page 96. The second training phase for DQS receiver enable is completed using the following procedure:

1. Select two test addresses for each chip select present in the system. The addresses must be cache line (64 byte) aligned and separated by 2Meg starting with the first rank.
2. Program F2x[1, 0]78[DqsRcvEnTrain]=1.
3. Program the total delay setting for each byte of each F2x[1, 0]9C_x[2B:10] register for the DIMM with the corresponding [The DRAM DQS Receiver Enable Timing Control Registers] F2x[1, 0]9C_x[2B:10] delay setting that passed for all ranks in phase one above.
4. Write a cache line to the first test address for each rank with the following data pattern:

   1234_5678_8765_4321h
   2345_6789_9876_5432h
   5938_5824_3049_6724h
   2449_0795_9993_8733h
   4038_5642_3846_5245h
   2943_2163_0506_7894h
   1234_9045_9872_3467h
   1238_7634_3458_7623h

5. Write a cache line to the second test address for each rank with the following data pattern:

   1234_5678_8765_4321h
   2345_6789_9876_5432h
   5938_5824_3049_6724h
   2449_0795_9993_8733h
   4038_5642_3846_5245h
   2943_2163_0506_7894h
   1234_9045_9872_3467h
   1238_7634_3458_7623h

6. For each channel:
   - For each rank:
     a. Write the current DQS receiver enable total delay settings for each byte lane in each F2x[1, 0]9C_x[2B:10] register for the current rank.
     b. Program F2x[1, 0]78[MaxRdLatency] with the current greatest value of F2x[1, 0]9C_x[2B:10]. See section 2.8.8.8.4 [Calculating MaxRdLatency] on page 98.
     c. Read the first test address for the rank and compare the data read with the written value from step 3 above.
     d. Reset the read pointer in the DRAM controller FIFO by writing the current corresponding DQS receiver enable delay settings for each byte lane in each corresponding F2x[1, 0]9C_x[2B:10] register.
     e. Read the second test address for the current rank and compare the data read with the expected

value from step 4 above.

f. Save the total delay F2x[1, 0]9C_x[2B:10] setting that passes for both read patterns to identify a range of passing values for each byte lane.

g. Increment the current total delay F2x[1, 0]9C_x[2B:10] setting by one for each byte lane in F2x[1, 0]9C_x[2B:10] and repeat steps a through f incrementing through all passing values for each byte lane until a fail setting is reached.

7. For each DIMM (chip select pair):
   For each total delay setting in each F2x[1, 0]9C_x[2B:10] register saved in phase 2, calculate the median value for the passing range obtained in phase 2, step #6 above. This centers the DQS receiver enable within the preamble and is the final total delay setting to be used for each byte lane in each F2x[1, 0]9C_x[2B:10] register.
8. Program F2x[1, 0]78[MaxRdLatency] with the current greatest value of F2x[1, 0]9C_x[2B:10] in phase 2, step 7 above. See section 2.8.8.8.4 [Calculating MaxRdLatency] on page 98.
9. Program F2x[1, 0]78[DqsRcvEnTrain]=0.

2.8.8.8.2.3 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[302:301, 202:201, 102:101, 02:01] and F2x[1, 0]9C_x[303, 203, 103, 03]) and the DRAM Read DQS Timing Control registers (see F2x[1, 0]9C_x[306:305, 206:205, 106:105, 06:05] and F2x[1, 0]9C_x[307, 207, 107, 07]) registers for DQS position training.

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.
To generate the needed continuous read or write data streams for DQS position training, see section 2.8.8.8.5 [Continuous Pattern Generation] on page 100.

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 byte lane

by two 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.

2.8.8.8.3 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. This affects the BIOS controlled training algorithms described above in sections 2.8.8.8.1.2 [BIOS Based Write Levelization Training] on page 89, 2.8.8.8.2.2 [BIOS Based DQS Receiver Enable Training] on page 94, and 2.8.8.8.2.3 [DQS Position Training] on page 96. 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 4 and 5, the programmed value for the delay registers would be the average of the values used for data byte lanes 4 and 5. 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, ECC delay training is accomplished by adding or subtracting a fixed delay value based on adjacent DRAM locations relative to the ECC DRAM on the DIMM. Currently, there are five families of DDR3 registered DIMMs with ECC: 1 rank x8, 2 rank x8, 1 rank x4, 2 rank x4 stacked card, and 2 rank x4 planar card. All data lanes on x4 DIMMs are trained using the lower nibbles by default so the data lanes chosen for the examples below are byte aligned. See 2.8.8.8.1 [DDR3 Training] on page 86. The delay settings

for the first four family types can be calculated using the method as described in the following example.

For each rank of a 1 or 2 rank x8 DDR3 registered DIMM, BIOS calculates the needed delay value by subtracting the delay value used for data lane X associated with DRAM device D2 from the value used for data lane Y associated with DRAM device D3. See Figure 6 below. The difference is added to the delay setting used for data lane Y and the result is used for the delay setting of the ECC lane. BIOS repeats this process for the other ranks as needed. A similar method is used for 2 stacked rank x4 registered DDR3 devices.

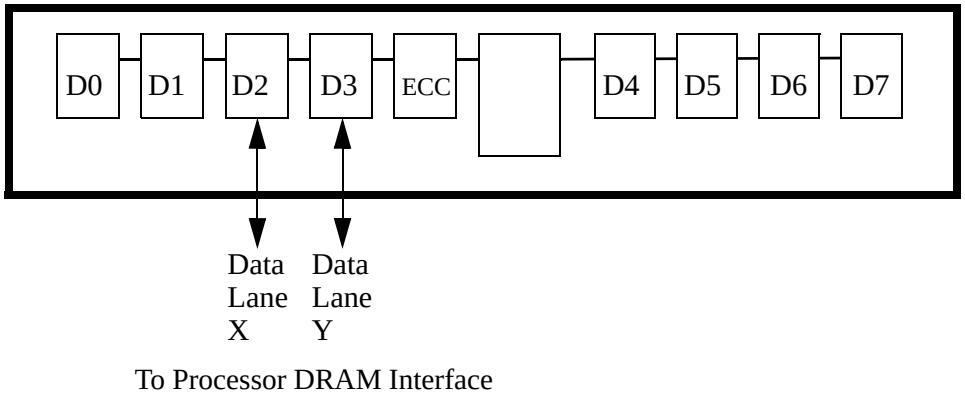

Figure 6: DDR3 Registered DIMM (non-planar)

For DDR3 planar registered DIMMs, ECC delay settings are determined by adding the difference in the delay settings between two adjacent DRAMs on a DIMM to the value used for the DRAM device next to the ECC DRAM device on the DIMM. This value is used as the delay setting for the ECC lane. For the x4 planar DIMM example shown in Figure 7, the delay values needed to calculate the ECC delay setting are based on byte aligned data lanes. In this case, BIOS calculates the needed delay value by subtracting the delay value used for data lane X associated with DRAM device D2 from the value used for data lane Y associated with DRAM device D9. See Figure 7 below. The difference is added to the delay setting used for data lane Y and the result is used for the delay setting of the ECC lane. BIOS repeats this process for the other ranks as needed.

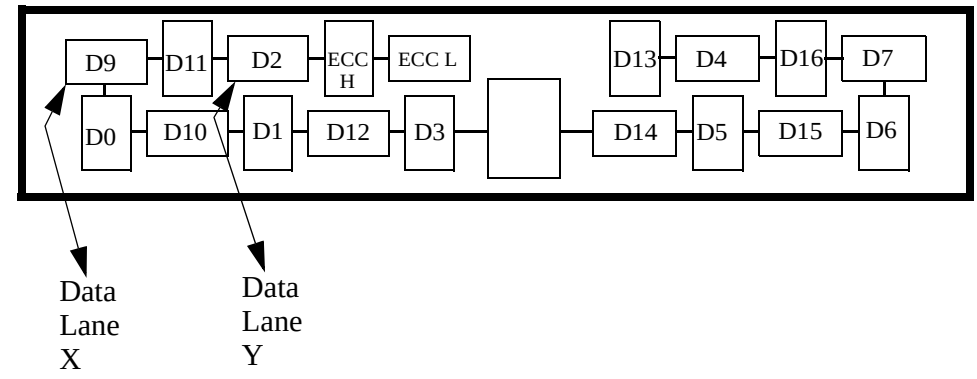

Figure 7: DDR3 Registered DIMM (planar)

2.8.8.8.4 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 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 Tcl and store into K.
   - K = 2 * CL; See F2x[1, 0]88[Tcl].
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 + (Maximum F2x[1, 0]9C_x[2B:10][DqsRcvEnGrossDelay])
6. Add 5.5 to the sub-total K. 5.5 represents part of the processor specific constant delay value in the DRAM clock domain.
   - K = K + 5.5
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 /(current memory clock frequency); see F2x[1, 0]88[MemClkFreq]
   - K = K / 2; removes the 1/2 MEMCLK component
8. Add 5 NCLKs to the sub-total. 5 represents part of the processor specific constant delay value in the Northbridge clock domain.
   - K = K + 5
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.

2.8.8.8.4.1 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_3FACh
49C5_B613_4A68_8181h
5C16_50E3_7C78_0BA6h
0C67_53E6_0C4F_9D76h
BABF_B6CA_2055_35A5h
0C5F_1C87_610E_6E5Fh
```

```
14C9_C383_4884_93CEh
9CE8_F615_F5B9_A5CDh

C38F_1B4C_AAD7_14B5h
669F_7562_72ED_647Ch
4A89_8B30_5233_F802h
3326_B465_10A4_0617h
C807_E3D3_5538_6E04h
14B4_E63A_AB49_E193h
EA51_7C45_67DF_2495h
F814_0C51_7624_CE51h

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

- For each channel:
  - BIOS calculates a starting MaxRdLatency delay value by executing steps 1 through 5, and 7 in section 2.8.8.8.4 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.
  - Using the patterns given above, write 3 cache lines to the target address on the current DIMM.
    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 section 2.8.8.8.4 step 7 above) for each channel. If the channels are ganged, use the larger value.

2.8.8.8.5 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 corre-

sponding 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 2.8.3 [Burst Length] on page 65. 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. BIOS generates two 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) 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 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 section 2.8.9.2 [Node Interleaving] on page 104.

Any combination of these interleaving modes may be enabled concurrently.

2.8.9.1 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 F2x[1, 0][5C:40][BaseAddr[36:27]] bits with F2x[1, 0][5C:40][BaseAddr[21:13]] bits, as defined Table 26 and Table 27 for DDR2 and Table 28 and Table 29 for DDR3.
3. For each enabled chip select, swap the corresponding F2x[1, 0][6C:60][AddrMask[36:27]] bits with F2x[1, 0][6C:60][AddrMask[21:13]] bits, as defined in Table 26 and Table 27 for DDR2 and Table 28 and Table 29 for DDR3.

Table 26. 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]
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 27. 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]

Table 27. 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
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 28. 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 29. 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]

Table 29. 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
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 DDR2 DRAM. The DRAM memory consists of two 2-sided DDR2 DIMMs with 256 Mbytes on each side.

1. 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_3FF0h // CS0/CS1 = 256 MB

   F2x[1, 0]64 = 0008_3FF0h // CS2/CS3 = 256 MB

2. The base address bits to be swapped are defined in Table 26, 256MB chip select size, 4 way CS interleaving column (4 chip selects are used). Base address bits [29:28] bits are defined with F2x[1, 0][5C:40][BaseAddr[21:20]. Base address bits [16:15] are defined with F2x[1, 0][5C:40][BaseAddr[8:7]].

   F2x[1, 0]40 = 0000_0001h

   F2x[1, 0]44 = 0000_0081h

   F2x[1, 0]48 = 0000_0101h

   F2x[1, 0]4C = 0000_0181h

3. The address mask bits to be swapped are the same as the base address bits defined in the previous step. Address mask bits [29:28] are defined with F2x[1, 0][6C:60][AddrMask[21:20]. Address mask bits [16:15] are defined with F2x[1, 0][6C:60][AddrMask[8:7].

   F2x[1, 0]60 = 0038_3E70h

   F2x[1, 0]64 = 0038_3E70h

2.8.9.2 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:

Node 0
- IntlvEn = 00_0011b
- IntlvSel = 000b
- Addr[13:12] = 00b

Node 1
- IntlvEn = 00_0011b
- IntlvSel = 001b
- Addr[13:12] = 01b

Node 2
- IntlvEn = 00_0011b
- IntlvSel = 010b
- Addr[13:12] = 10b

Node 3
- IntlvEn = 00_0011b
- IntlvSel = 011b
- Addr[13:12] = 11b

2.8.10 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: DramHoleSize[31:24] = (100h-DramHoleBase[31:24]).

2.8.10.1 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:
  DramHoleOffset[31:23] = {DramHoleSize[31:24], 0b} + {DramBaseAddr[31:27], 0000b};
- **Case 2**: if both DCTs are enabled in channel non-interleaved mode and DctSelBaseAddr < DramHoleBase, then:
  DramHoleOffset[31:23] = {DramHoleSize[31:24], 0b} + {DctSelBaseAddr[31:27], 0000b};
- **Case 3**: if both DRAM controllers are enabled in channel interleaved mode and DctSelBaseAddress < DramHoleBase, then:
  DramHoleOffset[31:23] = { DramHoleSize[31:24], 0b} + {DramBaseAddr[31:27], 0000b}
  + {0b, (DctSelBaseAddr[31:27] - DramBaseAddr[31:27]), 000b};

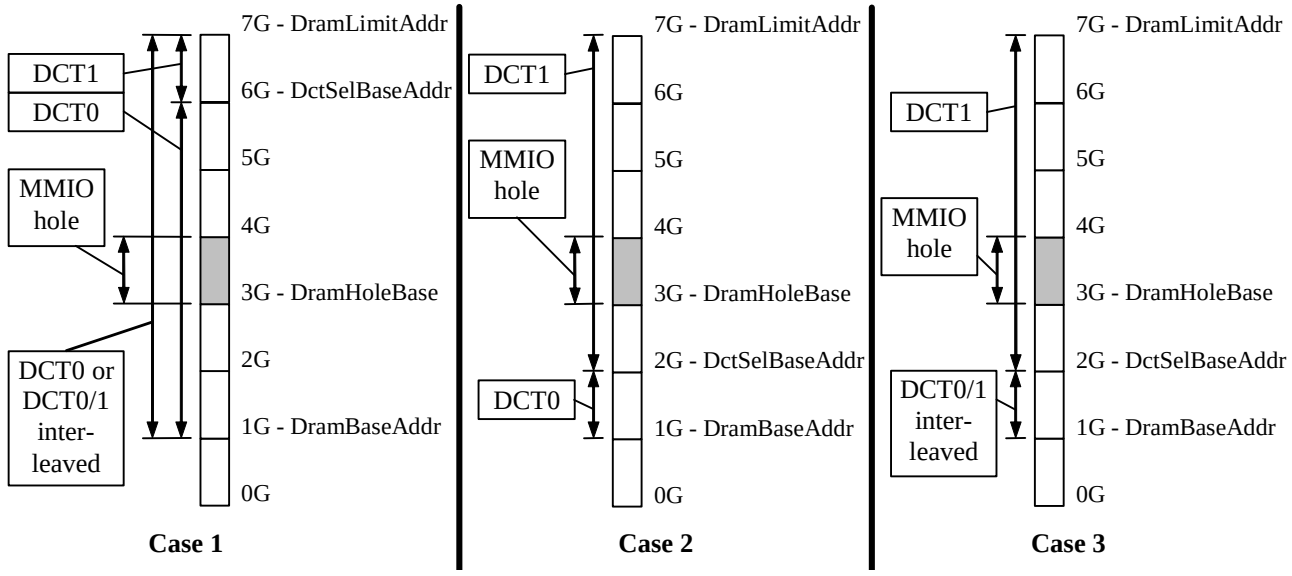

Figure 8: Example cases for programming DramHoleOffset.

2.8.10.2 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:
  DctSelBaseOffset[47:26] = {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:
  DctSelBaseOffset[47:26] = {DramBaseAddr[47:27], 0b}
  + {0b, (DctSelBaseAddr[47:27] – 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:
  DctSelBaseOffset[47:26] = {DramBaseAddr[47:27], 0b}
  + {0000h, DramHoleSize[31:26]}
  + {0b, (DctSelBaseAddr[47:27]
  – {0000h, (DramBaseAddr[31:27] + DramHoleSize[31:27]) } };

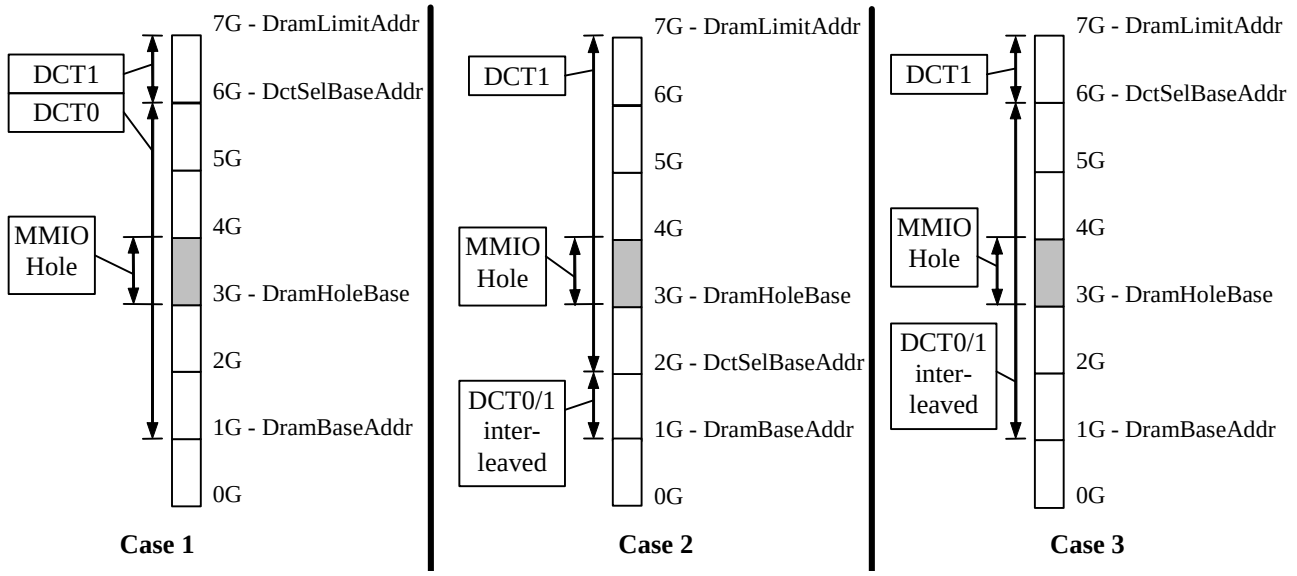

Figure 9: Example cases for programming DctSelBaseOffset.

2.8.11 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.

2.8.11.1 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.

2.9 CPU Core

The majority of the behavioral definition of the CPU core is specified in the *AMD64 Architecture Programmer's Manual*. See section 1.2 [Reference Documents] on page 13.

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, or 4 cores as follows:

- The number of cores supported by the node is specified by F3xE8[CmpCap].
- Cores may be *downcored* (removed) by F3x190[DisCore[3:0]] through a warm reset. This may be useful in that cores that are determined to be bad may be removed from operation. 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.
- F3x190[DisCore] affects CPUID Fn8000_0008_ECX[NC].
- Software is required to use F3x190[DisCore[3:0]] as follows:
  - 1, 2, 3 or 4 cores must be enabled on each node (0-core configurations are not allowed).
  - 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] in all nodes must be updated to reflect the new value after the warm reset.
- 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[Cpu3En and Cpu2En]. CpuCoreNum, varies as the lowest integers from 0 to 3, 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. Here are all the possible downcore combinations:
  - A 4-core node with 0 cores disabled.
  - A 4-core node with 1 core disabled.
  - A 4-core node with 2 cores disabled.
  - A 4-core node with 3 cores disabled.
  - A 1-core node with 0 cores disabled.
  - A 3-core node with 0 cores disabled.
  - A 3-core node with 1 core disabled.
  - A 3-core node with 2 cores disabled.
  - A 2-core node with 0 cores disabled.
  - A 2-core node with 1 core disabled.
- The boot core is always the core reporting CpuCoreNum=0.

Some legacy operating systems do not support three core processors. The BIOS should support a user configurable option to disable one core in a three 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] on page 54.

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 MTRRphysMaskn)] 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 Fixed-Size MTRRs (MTRRfixn)] MSR0000_02[6F:68, 59, 58, 50].
3. The IORRs (see MSRC001_00[18, 16] and MSRC001_00[19, 17]).
4. TSEG & ASEG (see MSRC001_0112 and MSRC001_0113).
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 guarantee 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 increments at the rate of CLKIN; the APIC timer may increment in units of between 1 and 8.

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. See2.16.1 [Multi-Core Support] on page 135 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:

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

Where OFFSET_IDX is an integer offset (0 to N) used to shift up the 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 = (Number_Of_Processors * MNC) and M = 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) > =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=0,core=1] = (2+0)*4 + 1 = 0x09
ApicId[proc=0,core=2] = (2+0)*4 + 2 = 0x0A
ApicId[proc=1,core=0] = (2+1)*4 + 0 = 0x0C
ApicId[proc=1,core=1] = (2+1)*4 + 1 = 0x0D
ApicId[proc=1,core=2] = (2+1)*4 + 2 = 0x0E
ApicId[proc=2,core=0] = (2+2)*4 + 0 = 0x10
ApicId[proc=2,core=1] = (2+2)*4 + 1 = 0x11
ApicId[proc=2,core=2] = (2+2)*4 + 2 = 0x12

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.

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 section 2.4.2 [P-states] on page 34. 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 Datasheet.

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 section 2.10.1 [The Tctl Temperature Scale] on page 111) 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 spec-

ified by [The Software Thermal Control (STC) Register] F3x68[StcPstateLimit]; See section 2.4.2 [P-states] on page 34.

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] on page 114 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] on page 56 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] on page 137. 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 Debug Support

2.13 . RAS and Advanced Server Features

This section applies reliability, availability, and serviceability, or *RAS*, and related advanced server considerations.

2.13.1 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 section 1.2 [Reference Documents] on page 13.

2.13.1.1 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 MC*i*_CTL register in each bank must be enabled by the corresponding enable bit in MCG_CTL (MSR0000_017B).
    - **MCi_STATUS**: The Machine Check Status Register: Logs information associated with errors.
    - **MCi_ADDR**: The Machine Check Address Register: Logs address information associated with errors.
    - **MCi_MISC**: The Machine Check Miscellaneous Registers: Log miscellaneous information associated with errors, as defined by each error type.
  - The six error-reporting register banks supported are:
    - **MC0, DC**: MSR0000_04[03:00], data cache machine check registers.
    - **MC1, IC**: MSR0000_04[07:04], instruction cache machine check registers.
    - **MC2, BU**: MSR0000_04[0B:08], bus unit machine check registers.
    - **MC3, LS**: MSR0000_04[0F:0C], load-store machine check registers.
    - **MC4, NB**: MSR0000_04[13:10], Northbridge machine check registers. 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 30 identifies the addresses associated with each MCA register.

Table 30: MCA register cross-reference table

Register Bank (MC*i*)	MCA Register				
	CTL	STATUS	ADDR	MISC	CTL_MASK MSRC001_00[49:44]
MC0	MSR0000_0400	MSR0000_0401	MSR0000_0402	MSR0000_0403	MSRC001_0044
MC1	MSR0000_0404	MSR0000_0405	MSR0000_0406	MSR0000_0407	MSRC001_0045
MC2	MSR0000_0408	MSR0000_0409	MSR0000_040A	MSR0000_040B	MSRC001_0046
MC3	MSR0000_040C	MSR0000_040D	MSR0000_040E	MSR0000_040F	MSRC001_0047
MC4	MSR0000_0410	MSR0000_0411	MSR0000_0412	MSR0000_0413 MSRC000_04[0A:08]	MSRC001_0048

Table 30: MCA register cross-reference table

Register Bank (MC*i*)	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 2.13.1.4 [Error Thresholding] on page 119).

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.

2.13.1.2 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.

2.13.1.2.1 Machine Check Error Logging and Reporting

An error is considered enabled for logging if:
- The global enable for the corresponding error-reporting bank in [The Global Machine Check Exception Reporting Control Register (MCG_CTL)] MSR0000_017B is set to 1.
- The corresponding mask bit for the error in [The Machine Check Control Mask Registers (MCi_CTL_MASK)] MSRC001_00[49:44] is cleared to 0.

An error is considered enabled for reporting if:
- The error is enabled for logging.
- The corresponding enable bit for the error in MC*i*_CTL is set to 1.

2.13.1.2.2 Machine Check Error Logging Overwrite During Overflow

During error overflow conditions (see MSR0000_0401[Over] and MSR0000_0411[Over]), an error which has already been logged in the status register may be overwritten.

Table 31 indicates which errors are overwritten in the MC0 and MC4 error status registers. Table 32 indicates which errors are overwritten in the MC1, MC2, MC3, and MC5 error status registers.

Table 31: 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 32: 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

2.13.1.3 Handling Machine Check Exceptions

At a minimum, the machine check handler must be capable of logging errors for later examination. The handler should log as much information as is needed to diagnose the error.

More thorough exception handler implementations can analyze errors to determine if each error is recoverable. If a recoverable error is identified, the exception handler can attempt to correct the error and restart the interrupted program. Keep in mind that an error may not be recoverable for the process it directly affects, but may be containable to only that process, so that other processes in the system are unaffected.

Machine check exception handlers that attempt to recover must be thorough in their analysis and the corrective actions they take. The following guidelines should be used when writing such a handler:

- All status registers in the error-reporting banks must be examined to identify the cause of the machine check exception. Read [The Global Machine Check Capabilities Register (MCG_CAP)] MSR0000_0179[Count] to determine the number of status registers visible to each core. The status registers are numbered from 0 to one less than the value found in MSR0000_0179[Count]. For example, if the Count field indicates five status registers are supported, they are numbered MC0_STATUS to MC4_STATUS.
- Check the valid bit in each status register (MC*i*_STATUS[Val]). The remainder of the MC*i*_STATUS register does not need to be examined when its valid bit is clear.
- When identifying the error condition, portable exception handlers should examine MC*i*_STATUS[Error Code] and [ErrorCodeExt].
- When logging errors, particularly those that are not recoverable, check [The Global Machine Check Status Register (MCG_STAT)] MSR0000_017A[EIPV] to see if the instruction pointer address pushed onto the exception handler stack is related to the machine check. If EIPV is clear, the address is not guaranteed to be related to the error.
- Check the valid MC*i*_STATUS registers to see if error recovery is possible. Error recovery is not possible when:
  - The processor context corrupt indicator (MC*i*_STATUS[PCC]) is set to 1.
  - The error overflow status indicator (MC*i*_STATUS[Over]) is set to 1. This indicates that more than one machine check error has occurred, but only one error is reported by the status register.

  If error recovery is not possible, the handler should log the error information and return to the operating system.
- Check MC*i*_STATUS[UC] to see if the processor corrected the error. If UC is set, the processor did not correct the error, and the exception handler must correct the error prior to attempting to restart the interrupted program. If the handler cannot correct the error, it should log the error information and return to the operating system.
- If [The Global Machine Check Status Register (MCG_STAT)] MSR0000_017A[RIPV] is set, the interrupted program can be restarted reliably at the instruction pointer address pushed onto the exception handler stack. If RIPV is clear, the interrupted program cannot be restarted reliably, although it may be possible to restart it for debugging purposes.
- Prior to exiting the machine check handler, be sure to clear [The Global Machine Check Status Register (MCG_STAT)] MSR0000_017A[MCIP]. MCIP indicates that a machine check exception is in progress. If this bit is set when another machine check exception occurs, the processor enters the shutdown state.
- When an exception handler is able to successfully log an error condition, clear the MC*i*_STATUS registers prior to exiting the machine check handler. Software is responsible for clearing at least MC*i*_STATUS[Val].

Additional machine check handler portability can be added by having the handler use the CPUID instruction to identify the processor and its capabilities. Implementation specific software can be added to the machine check

exception handler based on the processor information reported by CPUID.

2.13.1.4 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 indicate 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. Thresholding gives error counts on groups of resources. Whenever possible, a finer granularity of error information, such as MCA information for specific errors, should be utilized in order to obtain more accurate counts and limit the scope of actions to affected hardware.

Thresholding is performed for the following error threshold groups as identified in Table 47 on page 217. 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 2.8.11 [On-Line Spare] on page 107.
- Links
  - Link errors can be counted and reported via MSRC000_0408 (see MSRC000_04[0A:08]).
- L3 cache
  - L3 cache errors can be counted and reported via MSRC000_0409 (see MSRC000_04[0A:08]).

2.13.1.5 Scrub Rate Recommendations

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. This section gives guidelines for the scrub rate settings available in [The Scrub Rate Control Register] F3x58.

There are many factors which influence scrub rates. Among these are:

- The size of memory or cache to be scrubbed
- Resistance to upsets
- Geographic location and altitude
- Alpha particle contribution of packaging
- Performance sensitivity
- Risk aversion

The baseline recommendations which follow are intended to provide excellent protection at most geographic locations, while having no measurable effect on performance. Adjustments may be necessary due to special

circumstances.

- L1 cache: F3x58[DcacheScrub] baseline of 5.24 ms.
- L2 cache: F3x58[L2Scrub] baseline of 1.31 ms.
- L3 cache: F3x58[L3Scrub] baseline of 655.4 us.
- DRAM: F3x58[DramScrub] should be set to scrub all of memory every 6 to 12 hours, unless other guidelines are given by the DRAM vendor.

For steady state operation, finding a range of reasonable scrub rates is fairly straightforward; select 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. This allows a wide range of choices.

For low power states in which the processor core is halted, the power management configuration may affect scrubbing; see section 2.6.6 [Memory Scrubbers] on page 58 for details

2.13.1.6 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: F0x[14C: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.

2.13.2 DRAM Considerations for ECC

DRAM is protected against errors by an error correcting code (ECC). The DRAM error correcting code employed is a 128/16 (data bits/check bits) SSC-DSD (Single Symbol Correction, Double Symbol Detection) BCH code. A symbol is a group of 4 bits which are 4-bit aligned. Bits 0–3 make symbol 0, bits 4–7 make symbol 1, and so on.

A single symbol error is any bit error combination within one symbol. The ECC is 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 more than two symbol errors depending on the position of corrupted symbols.

ECC has different characteristics depending on the physical configuration of the memory, including DRAM device width, ganged vs. unganged DRAM modes, and data interleaving. DRAM device width refers to the number of bits sourced simultaneously from a single memory chip. 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. For a description of ganged (128-bit DRAM data width) and unganged (64-bit DRAM data width) DRAM modes, see section 2.8 [DRAM Controllers (DCTs)] on page 64. Data interleaving refers to the way bits from the different memory beats are organized to form an ECC line. For the proper setting of data interleaving while in unganged mode, see F2x110[DctDatIntLv]. For a more detailed description of interleaving, refer to section 2.13.2.1 [Unganged Interleaving] on page 121.

In certain configurations, the ECC provides “chipkill” functionality; all single symbol errors caused by a failed DRAM device are corrected. Chipkill recovery is only possible when indicated in F3x44[ChipKillEccEn] and the symbol size is greater than or equal to the DRAM device width. 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[ChipKillEccEn].

The configuration specific recommendations and detection and correction characteristics are as follows:

- For x4 DRAM devices:
  - In ganged mode, the failure of a DRAM device results in an error to a single symbol and can be corrected (chipkill).
  - In unganged mode, the failure of a DRAM device results in errors to two symbols, and can be detected but cannot be corrected.
- For DRAM devices wider than 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. Note that 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.
  - In ganged mode with x8 devices, failure of a x8 device can be detected with 100% probability, since exactly two symbols are affected.
  - In unganged mode or with devices wider than x8, device failures affect more than two symbols and have a lower probability of detection.

2.13.2.1 Unganged Interleaving

Note that in unganged DRAM mode (section 2.8 [DRAM Controllers (DCTs)] on page 64), even and odd bits from two 64-bit DRAM data beats are interleaved to create a 128-bit line as shown in Figure 10. The ECC function is applied to this 128-bit result. When using the ECC syndrome to find the bits in error, use Figure 10 to map from the symbol and bit number back to the correct device bit. Refer to 2.8.5 [Routing DRAM Requests] on page 66 for details on how to map to a DIMM and device.

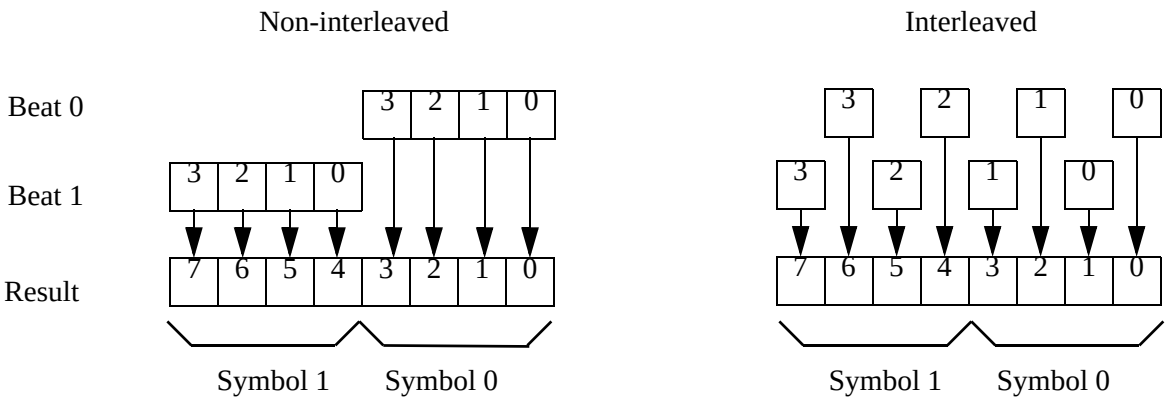

Figure 10: Example of line interleaving from x4 DRAM in unganged DRAM mode

When two lines are interleaved, 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.

2.13.2.2 ECC Syndromes

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.

The syndrome field uniquely identifies the failing bit positions of a correctable ECC error. Only syndromes identified by Table 33 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 33, 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 2.8.5 [Routing DRAM Requests] on page 66.

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 33: 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

Table 33: ECC correctable syndromes

Symbol In Error	Error Bitmask														
	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Data 16	45d1	8a62	cfb3	5e34	1be5	d456	9187	a718	e2c9	2d7a	68ab	f92c	bcfd	734e	369f
Data 17	63e1	b172	d293	14b4	7755	a5c6	c627	28d8	4b39	99aa	fa4b	3c6c	5f8d	8d1e	eeff
Data 18	b741	d982	6ec3	2254	9515	fbd6	4c97	33a8	84e9	ea2a	5d6b	11fc	a6bd	c87e	7f3f
Data 19	dd41	6682	bbc3	3554	e815	53d6	8e97	1aa8	c7e9	7c2a	a16b	2ffc	f2bd	497e	943f
Data 20	2bd1	3d62	16b3	4f34	64e5	7256	5987	8518	aec9	b87a	93ab	ca2c	e1fd	f74e	dc9f
Data 21	83c1	c142	4283	a4f4	2735	65b6	e677	f858	7b99	391a	badb	5cac	df6d	9dee	1e2f
Data 22	8fd1	c562	4ab3	a934	26e5	6c56	e387	fe18	71c9	3b7a	b4ab	572c	d8fd	924e	1d9f
Data 23	4791	89e2	ce73	5264	15f5	db86	9c17	a3b8	e429	2a5a	6dcb	f1dc	b64d	783e	3faf
Data 24	5781	a9c2	fe43	92a4	c525	3b66	6ce7	e3f8	b479	4a3a	1dbb	715c	26dd	d89e	8f1f
Data 25	bf41	d582	6ac3	2954	9615	fcd6	4397	3ea8	81e9	eb2a	546b	17fc	a8bd	c27e	7d3f
Data 26	9391	e1e2	7273	6464	f7f5	8586	1617	b8b8	2b29	595a	cacb	dcdc	4f4d	3d3e	aeaf
Data 27	cce1	4472	8893	fdb4	3155	b9c6	7527	56d8	9a39	12aa	de4b	ab6c	678d	ef1e	23ff
Data 28	a761	f9b2	5ed3	e214	4575	1ba6	bcc7	7328	d449	8a9a	2dfb	913c	365d	688e	cfef
Data 29	ff61	55b2	aad3	7914	8675	2ca6	d3c7	9e28	6149	cb9a	34fb	e73c	185d	b28e	4def
Data 30	5451	a8a2	fcf3	9694	c2c5	3e36	6a67	ebe8	bfb9	434a	171b	7d7c	292d	d5de	818f
Data 31	6fc1	b542	da83	19f4	7635	acb6	c377	2e58	4199	9b1a	f4db	37ac	586d	82ee	ed2f
Check0	be01	d702	6903	2104	9f05	f606	4807	3208	8c09	e50a	5b0b	130c	ad0d	c40e	7a0f
Check1	4101	8202	c303	5804	1905	da06	9b07	ac08	ed09	2e0a	6f0b	f40c	b50d	760e	370f
Check2	c441	4882	8cc3	f654	3215	bed6	7a97	5ba8	9fe9	132a	d76b	adfc	69bd	e57e	213f
Check3	7621	9b32	ed13	da44	ac65	4176	3757	6f88	19a9	f4ba	829b	b5cc	c3ed	2efe	58df

2.13.3 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 withthe Temperature Sensor Interface (SB-TSI) (see the *SBI Temperature Sensor Interface (SB-TSI) Specification*, #40821).

2.13.3.1 SBI Processor Information

Processor access to the SBI configuration is via [The SBI Control Register] F3x1E4. The processor can access SB-TSI registers through [The SBI Address Register] F3x1E8 and [The SBI Data Register] F3x1EC.

2.14 Interrupts

2.14.1 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:

- I/O devices including the I/O hub (I/O APICs)
- Other local APICs (inter-processor interrupts)
- APIC timer
- Thermal events
- Performance counters
- Legacy local interrupts from the I/O 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.

I/O 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.

2.14.1.1 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.

2.14.1.2 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.

2.14.1.3 Interrupt Delivery

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

When an APIC accepts a non-vectored interrupt, it is handled directly by the processor instead of being queued in the APIC. When an APIC accepts a fixed or lowest-priority interrupt, it sets the bit in [The Interrupt Request Registers] APIC[270:200] corresponding to the vector in the interrupt. For local interrupt sources, this comes from the vector field in that interrupt's local vector table entry. 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.

2.14.1.4 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.

2.14.1.5 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.

2.14.1.6 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.

2.14.1.6.1 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.

2.14.1.7 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 APIC[170:100][InServiceBits] (ISRVec). If APIC410[IerCap] is set the IRRVec and ISRVec are based off the highest enabled interrupt. The main priority bits 7-4 are compared as follows:

If (APIC80[Priority[7:4]] >= IRRVec[7:4] and APIC80[Priority[7:4]] > ISRVec[7:4])
Then APIC90[Priority] = APIC80[Priority]
Else if (IRRVec[7:4] > ISRVec[7:4]) APIC90[Priority] = {IRRVec[7:4],0h}
Else APIC90[Priority] = {ISRVect[7:4],0h}

2.14.1.8 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.

2.14.1.9 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.

2.14.1.10 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

2.14.1.11 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.

2.14.2 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.

2.14.2.1 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.

2.14.2.2 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.

2.14.2.3 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.

2.14.2.4 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 34: 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

Table 34: SMM initial state

Register	SMM Initial State
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.

2.14.2.5 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 35: 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} (see note 1 at the end of this table)	
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} (see note 1 at the end of this table)	
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	

Table 35: SMM Save State

Offset	Size	Contents		Access
FE70h	Word	LDTR	Selector	Read-only
FE72h	Word		Attributes	
FE74h	Doubleword		Limit	
FE78h	Quadword		Base	
FE80h	4 Bytes	IDTR	reserved	Read-only
FE84h	Word		Limit	
FEB6h	2 Bytes		reserved	
FE88h	Quadword		Base	
FE90h	Word	TR	Selector	Read-only
FE92h	Word		Attributes	
FE94h	Doubleword		Limit	
FE98h	Quadword		Base	
FEA0h	Quadword	IO_RESTART_RIP		Read-only
FEA8h	Quadword	IO_RESTART_RCX		
FEB0h	Quadword	IO_RESTART_RSI		
FEB8h	Quadword	IO_RESTART_RDI		
FEC0h	Doubleword	[The SMM IO Trap Offset] SMMFEC0		Read-only
FEC4	Doubleword	[The Local SMI Status] SMMFEC4		Read-only
FEC8h	Byte	[The SMM IO Restart Byte] SMMFEC8		Read-write
FEC9h	Byte	[The Auto Halt Restart Offset] SMMFEC9		Read-write
FECAh	Byte	[The NMI Mask] SMMFECA		Read-write
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		
FF30h	Quadword	Host CR4		
FF38h	Quadword	Host CR3		
FF40h	Quadword	Host Cr0		
FF48h	Quadword	CR4		
FF50h	Quadword	CR3		
FF58h	Quadword	CR0		

Table 35: SMM Save State

Offset	Size	Contents	Access
FF60h	Quadword	DR7	Read-only
FF68h	Quadword	DR6	
FF70h	Quadword	RFLAGS	Read-write
FF78h	Quadword	RIP	Read-write
FF80h	Quadword	R15	
FF88h	Quadword	R14	
FF90h	Quadword	R13	
FF98h	Quadword	R12	
FFA0h	Quadword	R11	
FFA8h	Quadword	R10	
FFB0h	Quadword	R9	
FFB8h	Quadword	R8	
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	

Note 1: this notation specifies that bit[47] is replicated in each of the 16 MSBs of the DW (sometimes called *sign extended*). The 16 LSBs contain bits[47:32].

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

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

SMMFEC0 SMM IO Trap Offset

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

Bits	Description
31:16	**Port: trapped IO port address**. Read-only. This provides the address of the IO instruction.
15:12	**BPR: IO breakpoint match**. Read-only.

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

SMMFEC4 Local SMI Status

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

Bits	Description
31:23	Reserved.
22	**SmiSrcOnLineSpare: SMI source on-line spare**. This bit is associated with the SMI sources specified in [The On-Line Spare Control Register] F3xB0.
21	Reserved.
20	**SmiSrcThrCntL3: SMI source L3 cache thresholding**. 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	**SmiSrcThrCntHt: SMI source link thresholding**. 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	**SmiSrcThrCntDram: SMI source DRAM thresholding**. 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	**SmiSrcLvtExt: SMI source LVT extended entry**. This bit is associated with the SMI sources specified in [The Extended Interrupt [3:0] Local Vector Table Registers] APIC[530:500].
16	**SmiSrcLvtLcy: SMI source LVT legacy entry**. This bit is associated with the SMI sources specified by the non-extended LVT entries of the APIC.
15:11	Reserved.
10	**IntPendSmiSts: interrupt pending SMI status**. 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	**SmiOnCmpHaltSts: SMI on CMP halt status**. This bit is associated with the SMI source specified in [The Interrupt Pending and CMP-Halt Register] MSRC001_0055[SmiOnCmpHalt].
8	**MceRedirSts: machine check exception redirection status**. This bit is associated with the SMI source specified in [The Machine Check Exception Redirection Register] MSRC001_0022[RedirSmiEn].

7:4	Reserved.
3:0	**IoTrapSts: IO trap status**. Each of these bits is associated with each of the respective SMI sources specified in [The IO Trap Registers (SMI_ON_IO_TRAP_[3:0])] MSRC001_00[53:50].

SMMFEC8 SMM IO Restart Byte

00h on entry into SMM.

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

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

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

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

SMMFEC9 Auto Halt Restart Offset

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

SMMFECA NMI Mask

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

SMMFED8 SMM SVM State

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

Bits	Description
7:0	**SVM State**. Read-only. 00h=SMM entered from a non-guest state. 02h=SMM entered from a guest state. 06h=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	**BRL**. Read-only. Base relocation supported.
16	**IOTrap**. Read-only. IO trap supported.
15:0	**Revision**. Read-only.

SMMFF00 SMM Base Address Register (SMM_BASE)

This offset is loaded with the contents of MSRC001_0111. See that register for more details.

2.14.2.6 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.

2.14.2.7 The Protected ASeg and TSeg Areas

These ranges are controlled by MSRC001_0112 and MSRC001_0113; see those registers for details.

2.14.2.8 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].

2.14.2.9 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.

2.15 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.

2.15.1 BIOS support for SVM Disable

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

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

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 value supplied by user. This value should be NVRAM.

2.16 CPUID Instruction

The CPUID instruction provides data about the features supported by the processor. See section 3.9 [CPUID Instruction Registers] on page 285 for details.

2.16.1 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].

2.16.2 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[31:16] 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.

2.17 Performance Monitoring

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

2.17.1 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] on page 339.

2.17.2 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. The following performance information (see 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 instruction. 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] on page 113, 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 section 3.9 [CPUID Instruction Registers] on page 285. 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 36: Terminology in register descriptions

Terminology	Description
Read or read-only	Capable of being read by software. Read-only implies that the register cannot be 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, cleared-by-hardware, updated-by-hardware	Register bit is set high or cleared low 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 36: 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.
SBZ	Should be zero. If software attempts to set an SBZ bit to 1, it results in undefined behavior.
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 section 3.1 [Register Descriptions and Mnemonics] on page 137 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 also section 2.11 [Configuration Space] on page 113.

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	**ConfigEn: configuration space enable**. Read-write. 1=IO read and write accesses to IOCFC are translated into configuration cycles at the configuration address specified by this register. 0=IO read and write accesses are passed to the appropriate IO link and no configuration access is generated.
30:28	Reserved.
27:24	**ExtRegNo: extended register number**. Read-write. ExtRegNo provides bits[11:8] and RegNo provides bits[7:2] of the byte address of the configuration register. ExtRegNo is reserved unless it is enabled by MSRC001_001F[EnableCf8ExtCfg].
23:16	**BusNo: bus number**. Read-write. Specifies the bus number of the configuration cycle.
15:11	**Device: bus number**. Read-write. Specifies the device number of the configuration cycle.
10:8	**Function**. Read-write. Specifies the function number of the configuration cycle.
7:2	**RegNo: register address**. Read-write. See IOCF8[ExtRegNo].
1:0	Reserved.

IOCFC IO-Space Configuration Data Port

Reset: 0000 0000h.

Bits	Description
31:0	See IOCF8 for details about this port.

3.3 Function 0 HyperTransport™ Technology Configuration Registers

See section 3.1 [Register Descriptions and Mnemonics] on page 137 for a description of the register naming convention. See section 2.11 [Configuration Space] on page 113 for details about how to access this space.

F0x00 Device/Vendor ID Register

Reset: 1200 1022h.

Bits	Description
31:16	**DeviceID: device ID**. Read-only.
15:0	**VendorID: vendor ID**. Read-only.

F0x04 Status/Command Register

Reset: 0010 0000h.

Bits	Description
31:16	**Status**. Read-only. Bit[20] is set to indicate the existence of a PCI-defined capability block.
15:0	**Command**. Read-only.

F0x08 Class Code/Revision ID Register

Reset: 0600 0000h.

Bits	Description
31:8	**ClassCode**. Read-only. Provides the host bridge class code as defined in the PCI specification.
7:0	**RevID: revision ID**. Read-only.

F0x0C Header Type Register

Reset: 0080 0000h.

Bits	Description
31:0	**HeaderTypeReg**. Read-only. These bits are fixed at their default values. The header type field indicates that there are multiple functions present in this device.

F0x34 Capabilities Pointer Register

Reset: 0000 00??h.

Bits	Description
31:8	Reserved.
7:0	**CapPtr: capabilities pointer**. 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	**BCRoute: broadcast route**. Read-write. Specifies the routing information for broadcasts and probes.
17:9	**RPRoute: response route**. Read-write. Specifies the routing information for responses.
8:0	**RQRoute: request route**. 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	**CpuCnt[4:0]: CPU count bits[4:0]**. Read-write. This field specifies the number of cores to be enabled in the system (the boot core of all nodes plus those cores enabled through F0x68[Cpu1En] and F0x168[Cpu3En, Cpu2En]). 00h = 1 CPU... 1Fh = 32 CPUs. This field matches F0x60[NodeCnt] if each nodes in the system has one core; otherwise, it would be greater than F0x60[NodeCnt].
15	Reserved.
14:12	**LkNode[2:0]: lock node ID bits[2:0]**. Read-write. This field specifies the node ID of the node that contains the lock controller.
11	Reserved.
10:8	**SbNode[2:0]: Southbridge (IO Hub) node ID bits[2:0]**. Read-write. Specifies the node ID of the node that owns the link that connects to the system IO Hub.
7	Reserved.
6:4	**NodeCnt[2:0]: node count bits[2:0]**. 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 [The Northbridge Capabilities Register] F3xE8[MpCap]. 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	**NodeId[2:0]: node ID bits[2:0]**. 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 07h 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 also MSRC001_001F[InitApicIdCpuIdLo].

F0x64 Unit ID Register

Reset: 0000 00E0h.

Bits	Description
31:11	Reserved.

10:8	**SbLink: Southbridge (IO Hub) link ID**. 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 F0x60[NodeId]. 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	**HbUnit: host bridge Unit ID**. Read-only. This field specifies the coherent link Unit ID of the host bridge used by the coherent fabric.
5:4	**MctUnit: memory controller Unit ID**. Read-only. This field specifies the coherent link Unit ID of the memory controller.
3:2	Reserved.
1:0	**CpuUnit: CPU Unit ID**. Read-only. This field specifies the coherent link Unit ID used for CPU core transactions.

F0x68 Link Transaction Control Register

Reset: 0000 0000h.

Bits	Description
31	**EnPReqHiPriTblWlk: isoc table walk enable for posted requests**. 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 F3x1[54, 50, 4C, 48][IsocReqTok] must be allocated on each link that can receive DEV/GART table walk requests, and one F3x1[54, 50, 4C, 48][IsocRspTok] must be allocated on each link that can receive DEV/GART table walk responses.
30:26	Reserved.
25	**CHtExtAddrEn: coherent link extended address enable**. Read-write; however this bit is read-only, 0, for uniprocessor systems as indicated by F3xE8[MpCap]. 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	**DispRefModeEn**. 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 section 2.6.4.2.4 [F0x[5C:40]Display Refresh And IFCM] on page 57.
23	**InstallStateS**. Read-write. 1=Forces the default read block (RdBlk) install state to be shared instead of exclusive.
22:21	**DsNpReqLmt: downstream non-posted request limit**. Read-write. 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	**SeqIdSrcNodeEn: sequence ID source node enable**. 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	**ApicExtSpur: APIC extended spurious vector enable**. 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	**ApicExtId: APIC extended ID enable**. Read-write. This enables the extended APIC ID functionality. 0=APIC ID is 4 bits. 1=APIC ID is 8 bits.
17	**ApicExtBrdCst: APIC extended broadcast enable**. Read-write. This enables the extended APIC broadcast functionality. 0=APIC broadcast is 0Fh. 1=APIC broadcast is FFh.
16	**LintEn: local interrupt conversion enable**. Read-write. 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	**LimitCldtCfg: limit coherent link configuration space range**. 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	**BufRelPri: buffer release priority select**. 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. It is recommended that this be set to a value of 10b in order to maximize link bandwidth.
12	Reserved. Read-write.
11	**RespPassPW: response PassPW**. 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	**DisFillP: disable fill probe**. 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	**DisRmtPMemC: disable remote probe memory cancel**. 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	**DisPMemC: disable probe memory cancel**. 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	**CPURdRspPassPW: CPU read response PassPW**. 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	**CPUReqPassPW: CPU request PassPW**. 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	**Cpu1En: CPU core 1 enable**. Read-write. This bit and F0x168[Cpu3En and Cpu2En] 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 Cpu-CoreNum in section [The CPU Cores and Downcoring] 2.9.2.
4	**DisMTS: disable memory controller target start**. 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	**DisWrDwP: disable write doubleword probes**. 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	**DisWrBP: disable write byte probes**. 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	**DisRdDwP: disable read doubleword probe**. 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	**DisRdBP: disable read byte probe**. 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
31:20	Reserved.
19:16	Must be zero. Read-write.
15:12	Reserved.
11	**DefLnk[2]: default link**. Read-only. See DefLnk[1:0], below.
10:9	**BiosRstDet[2:1]: BIOS reset detect bits[2:1]**. Read-write. Cold reset: 0. See bit[5] of this register.
8	**DefSubLnk: default sublink**. Read-only. Used in conjunction with F0x6C[DefLnk]. 0=Sublink 0. 1=Sublink 1.
7	Reserved.
6	**InitDet: CPU initialization command detect**. 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	**BiosRstDet[0]: BIOS reset detect bit[0]**. 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	**ColdRstDet: cold reset detect**. Read-write. Cold reset: 0. This bit may be used to distinguish between a cold versus a warm reset event by setting the bit to a 1 before an initialization event is generated.
3:2	**DefLnk[1:0]: default link**. 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. 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. 100bRequest came from a core on same node.
1	**ReqDis: request disable**. 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 section 2.3 [Processor Initialization] on page 23.
0	**RouteTblDis: routing table disable**. 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] F0x[5C: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	**CapType: capability type**. Read-only, 001b.
28	**DropOnUnInit: drop on uninitialized link**. Read-only, 0.
27	**InbndEocErr: inbound end-of-chain error**. Read-only, 0.
26	**ActAsSlave: act as slave**. Read-only, 0.
25	Reserved.
24	**HostHide**. Read-only, 1.
23	**ChainSide**. Read-only, 0.
22:18	**DevNum: device number**. Read-only, 00h.
17	**DblEnded: double ended**. Read-only, 0.
16	**WarmReset**. Read-only, 1.

15:8	**CapPtr: capabilities pointer**. 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	**CapID: capabilities ID**. 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	**WidthOut: link width out**. Read-write. Cold reset: (see text below). Specifies the operating width of the outgoing link. Legal values are: Bits / Link width / Bits / Link width 001b / 16 bits / 111b / not connected 000b / 8 bits 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.
27	Reserved.
26:24	**WidthIn: link width in**. 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.
23	Reserved.
22:20	**MaxWidthOut: max link width out**. 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.
19	Reserved.
18:16	**MaxWidthIn: max link width in**. Read-only. This specifies the width of the incoming 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.

15	**Addr64BitEn: 64-bit address packet enable**. 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: F0x68[CHtExtAddrEn] 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 *warm* reset; therefore this bit may be in a different state than an IO device on the other side of the link after a warm reset; care should be taken by BIOS to place devices on both sides of the link in the same state after a warm reset, before any packets to the high-order addresses enabled by this bit are generated.
14	**ExtCTL: extended control time during initialization**. 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	**LdtStopTriEn: LDTSTOP tristate enable**. 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.
12	**IsocEn: isochronous flow-control mode enable**. 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 section 2.6.4.2.4 [F0x[5C:40]Display Refresh And IFCM] on page 57.
11:10	Reserved.
9:8	**CrcErr: CRC Error**. 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: • Link reconnect fails exceeding the limit in [The Link Global Retry Control Register] F0x150[TotalRetryAttempts].
7:6	Reserved.
5	**InitComplete: initialization complete**. 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.
4	**LinkFail: link failure**. 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.
3	**CrcForceErr: CRC force error command**. 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 also F0x150[ForceErrType] for retry mode.

2	Reserved.
1	**CrcFloodEn: CRC flood enable**. 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.
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	**LnkFreqCap: link frequency capability**. Read-only. Reset: values vary with product. These bits 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: Bit 0: 200 MHz (this bit is 1 in all products). Bit 1: 300 MHz (this bit is 0 in all products). Bit 2: 400 MHz. Bit 3: 500 MHz (this bit is 0 in all products). Bit 4: 600 MHz. Bit 5: 800 MHz. Bit 6: 1000 MHz. Bit 7: 1200 MHz. Bit 8: 1400 MHz. Bit 9: 1600 MHz. Bit 10: 1800 MHz. Bit 11: 2000 MHz. Bit 12: 2200 MHz. Bit 13: 2400 MHz. Bit 14: 2600 MHz. Bit 15: 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.
15:12	Reserved.
11:8	**Freq: link frequency**. Read-write. Cold reset: 0h. This specifies the link frequency. Legal values are: 0h: 200 MHz. 1h: reserved. 2h: 400 MHz. 3h: reserved. 4h: 600 MHz. 5h: 800 MHz. 6h: 1000 MHz. 7h: 1200 MHz. 8h: 1400 MHz. 9h: 1600 MHz. Ah: 1800 MHz. Bh: 2000 MHz. Ch: 2200 MHz. Dh: 2400 MHz. Eh: 2600 MHz. Fh: reserved. 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.
7:0	**Revision**. Read-only, 60h. Indicates that the processor is designed to version 3.00 of the link specification.

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	**UpstrCfgCap: upstream configuration capable**. Read-only, 0.
8	**ExtRegSet: extended register set**. Read-only, 0.
7:6	Reserved.
5	**UnitIdReOrderDis: UnitID reorder disable**. 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	**64BitAddr: 64-bit link addressing**. Read-only, 1.
3	**ExtCTLRqd: extended CTL required**. Read-only, 0.
2	**CrcTstMode: CRC test mode**. Read-only, 0.
1	**LdtStopMode: LDTSTOP supported**. Read-only, 1.
0	**IsocMode: isochronous flow control mode**. 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.

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 the whether the link initializes to IO or coherent protocol, ganged or unganged, as follows (if LockBc is low):

	FreeData	FreeCmd	RspData	NpReqData	ProbeCmd	RspCmd	PReq	NpReqCmd
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	16	1	1	0	2	4	18
Coh link unganged:	4	16	2	2	9	9	2	4

The cold-reset register state (ganged or unganged) is: IO link=0485_0292h; coherent link=048A_9944h.

For all fields except for FreeData and FreeCmd, 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 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 (even if LockBc is set).
- The total number of command buffers allocated in the base and isochronous registers of a link cannot exceed 64.
- The total number of data buffers allocated in the base and isochronous registers of a link cannot exceed 16.
- The total number of hard allocated command buffers (ProbeCmd, RspCmd, PReq, and NpReqCmd) cannot exceed 48.
- If unganged, the total buffer counts of a link are shared between the two sublinks.
- If unganged, 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].
- 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 the system is a UMA system and F0x84[IsocEn]=1, the following link buffer allocations should be used:

FreeData = 4	FreeCmd = 8	RspData = 1
NpReqData = 1	ProbeCmd = 0	RspCmd = 2
PReq = 3	NpReqCmd = 11	F0x94[IsocRspData] = 0
F0x94[IsocNpReqData] = 0	F0x94[IsocRspCmd] = 0	F0x94[IsocPReq] = 1
F0x94[IsocNpReqCmd] = 7		

Bits	Description
31	**LockBc: lock buffer count register**. Read-write. Cold reset: 0. 1=The buffer count registers, F0x[F0, D0, B0, 90] and F0x[F4, D4, B4, 94] 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.
30:28	Reserved
27:25	**FreeData: free data buffer count**. Read-write.
24:20	**FreeCmd: free command buffer count**. Read-write.
19:18	**RspData: response data buffer count**. Read-write.
17:16	**NpReqData: non-posted request data buffer count**. Read-write.
15:12	**ProbeCmd: probe command buffer count**. Read-write.
11:8	**RspCmd: response command buffer count**. Read-write.

7:5	**PReq: posted request command and data buffer count**. Read-write. This specifies the number of posted command and posted data buffers allocated.
4:0	**NpReqCmd: non-posted request command buffer count**. 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.

Bits	Description
31:29	Reserved.
28:27	**IsocRspData: isochronous response data buffer count**. Read-write.
26:25	**IsocNpReqData: isochronous non-posted request data buffer count**. Read-write.
24:22	**IsocRspCmd: isochronous response command buffer count**. Read-write.
21:19	**IsocPReq: isochronous posted request command and data buffer count**. Read-write. This specifies the number of isochronous posted command and posted data buffers allocated.
18:16	**IsocNpReqCmd: isochronous non-posted request command buffer count**. Read-write.
15:8	**SecBusNum: secondary bus number**. 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 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.

Bits	Description
31:5	Reserved.
4	**LinkConPend: link connect pending**. 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 LinkCon bit.
3	Reserved.
2	**NC: non coherent**. Read-only. This bit specifies the link type. 0=coherent link. 1=IO link.
1	**InitComplete: initialization complete**. 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	**LinkCon: link connected**. Read-only. 1=The link is connected to another device. 0=The link is not connected. This is not valid until LinkConPend=0.

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	**ClumpEn**. 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	**RetryCount**. 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	**DataCorruptOut: sent corrupted data**. 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	**InitFail**. Read; write-1-to-clear. Cold reset: 0. 1=Initialization sequence failed on a link reconnect.
10	**StompedPktDet: stomped packet detected by receiver**. Read; write-1-to-clear. Cold reset: 0.
9	**RetryCountRollover**. Read; write-1-to-clear. Cold reset: 0. See RetryCount.

8	**RetryErrorDet: retry error detected**. Read; write-1-to-clear. Cold reset: 0. 1=A retry was initiated in one of the ways listed in RetryCount.
7:6	**ShortRetryAttempts**. 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	**DisRetryDataError: disable link retry on data packet error**. 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	**DisRetryAnyError: disable link retry on any packet error**. 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	**ForceRetryError**. 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 also [The Link Global Retry Control Register] F0x150[MultRetryErr].
0	**RetryModeEnable**. Read-write; changes take effect on next warm reset. Cold reset: 0. 1=Place the link in error retry mode when reconnecting after the next warm reset.

F0x150 Link Global Retry Control Register

Cold reset: 0007 0000h if F3xE8[LnkRtryCap]=1, 0000 0000h 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).

Bits	Description
31:19	Reserved.
18:16	**TotalRetryAttempts**. Read-write. 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	**HtRetryCrcDatInsDynEn: link retry CRC data insertion enable**. Read-write. 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.
12	**HtRetryCrcCmdPackDynEn: link retry CRC command packet dynamic mode enable**. Read-write. 1=Enables dynamic mode for CRC command packing on a coherent link. In this mode, command packing is suspended when a link not busy.

11:9	**HtRetryCrcDatIns: link retry CRC data insertion**. Read-write. 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
8	**HtRetryCrcCmdPack: link retry CRC command packing**. Read-write. 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.
7	Reserved.
6:5	**ForceErrType: force error type**. Read-write. 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	**MultRetryErr: multiple retry force error**. Read-write. 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.

F0x164 Coherent Link Traffic Distribution Register

Reset: 0000 0000h. See 2.6.4.2.3 [Link Traffic Distribution] on page 57 for details about link traffic distribution.

Bits	Description
31:24	Reserved.
23:16	**DstLnk[7:0]: distribution destination link**. 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	**DstNode[2:0]: coherent link distribution destination node**. Read-write. For cHTReqDistEn and cHTRspDistEn, DstNode[2:0] 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[2:0]. 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	**cHTPrbDistEn: coherent link probe distribution enable**. Read-write. Enables coherent link traffic distribution for the probe virtual channel. The probes affected by this bit are limited to those sourced from the local node (as opposed to being forwarded from another node).
1	**cHTRspDistEn: coherent link response distribution enable**. 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	**cHTReqDistEn: coherent link request distribution enable**. 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 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	**DisNcHtCmdThrottle: disable IO link command throttling**. 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	**ExtMmioMapAddSel: extended MMIO map address select**. 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.
7:2	Reserved.
1	**Cpu3En: CPU core 3 enable**. Read-write. See F0x68[Cpu1En].
0	**Cpu2En: CPU core 2 enable**. 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
31:17	Reserved.
16	**ImmUpdate: immediate update**. Read-write. Cold reset: 0. Many of the link phy registers, accessed through F4x1[98, 90, 88, 80], control electrical parameters that are unsafe to change while the link is operational; so the updates to these registers are normally withheld until the link is disconnected. However, under some (testing and characterization) circumstances, it is preferable to allow these changes to occur immediately, while the link is operational. ImmUpdate provides this option. 0=Writes to most of the link phy registers do not take effect in the link phy until the next LDTSTOP or warm reset disconnect. Reads from a link phy register after a write return the current value not the value pending until the link disconnects and reconnects. 1=Writes to the link phy registers are passed to the phy immediately. Reads always returns the value from the most recent write.
15:13	**ForceFullT0: force full T0 training time**. Read-write. Cold reset: 000b. This specifies the period of time that the link can be disconnected for an LDTSTOP before the full T0 training period is invoked. The time is measured approximately from the assertion of LDTSTOP_L until training 0 is about to start after LDTSTOP deassertion. If less than the specified time has expired, then training 0 specified by T0Time is used. BIOS should set this to 110b. The bits are encoded as follows: 000b = Always use T0Time. 001b = 400 microseconds. 010b = 800 microseconds. 011b = 1.6 milliseconds. 100b = 3.2 milliseconds. 101b = 6.4 milliseconds. 110b = 12.8 milliseconds. 111b = 25.6 milliseconds.
12:10	Reserved.
9	**RXCalEn: receiver calibration enable**. Read-write. Cold reset: 0. 1=Enable receiver offset calibration during all training 1 periods. BIOS should leave this bit in the cold reset state.
8	Reserved.
7:6	**InLnSt: inactive lane state**. Read-write. Cold reset: 00b. Specifies the state of inactive lanes of ganged links at Gen3 frequencies as follows: 00b= Same as warm reset except CAD is logical 0. 01b= Same as PHY OFF. 10b= Same as operational; CTL and CAD transmit undefined scrambled data. This encoding is only supported with 8- or 16-bit links. 11b= Same as disconnected per F0x[18C:170][LS2En]. Updates to this bit take effect on warm reset and LDTSTOP.
5:0	**T0Time: training 0 time**. Read-write. Cold reset: 3Ah. Specifies the amount of time to spend in training 0 when exiting the disconnected state. See also ForceFullT0, F4x1[9C, 94, 8C, 84]_x[530A, 520A][Ls2ExitTime], and section 2.7.6 [Link LDTSTOP_L Disconnect-Reconnect] on page 61. If T0Time[5:4]=00b, then the time = T0Time[3:0] * 0.1 usec (ranging from 0.0 to 1.5 usec). If T0Time[5:4]=01b, then the time = T0Time[3:0] * 0.5 usec (ranging from 0.0 to 7.5 usec). If T0Time[5:4]=10b, then the time = T0Time[3:0] * 2.0 usec (ranging from 0.0 to 30 usec). If T0Time[5:4]=11b and T0Time[3:0] ranges from 0h to Ah, then the time = T0Time[3:0] * 20 usec (ranging from 0.0 to 200 usec). If T0Time[5:4]=11b and T0Time[3:0] ranges from Bh to Fh, these values are reserved. BIOS should set T0Time according to the T0 training time requirement for the links' far-side receiver phase recovery time as determined by characterization.

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	**LaneSel: lanes select**. Read-write. Cold reset: 00b. This field only exists in the sublink 0 registers; in the sublink 1 registers, these bits are reserved. For unganged 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: Bits 16-bit link 8-bit link 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]} Bits 4-bit link 2-bit link 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 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.
11	**ILMEn: internal loopback mode (ILM) enable**. Read-write. Cold reset: 0. 1=ILM enabled. F4x1[9C, 94, 8C, 84]_x[DF, CF][XmtRdPtr and RcvRdPtr] must be 0 (the default) when ILM mode is used.
10	**BistEn: built-in self test (BIST) enable**. Read-write. Cold reset: 0. 1=The link BIST engine is enabled.
9	Reserved
8	**LS2En: LDTSTOP mode 2 enable**. Read-write. Cold reset: 0. 0=Use LS1 mode for power reduction when the link is disconnected. 1=Use LS2 mode.
7:4	Reserved.

3	**ScrambleEn: scrambling enable**. Read-write. Cold reset: 0b. 1=Scrambling enable. Updates to this bit take effect on warm reset and LDTSTOP. Software must clear this bit when transitioning from Gen3 to Gen1 protocol.
2:1	Reserved.
0	**Ganged**. Read-write; read-only 1 if the bit corresponding to the link in F3xE8[UnGangEn] is 0. 0=The link is unganged; 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 section 2.7.1.1 [Ganging And Unganging] on page 59). 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.

F0x1A0 Link Initialization Status Register

Bits	Description
31	**InitStatusValid: initialization status valid**. 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.
30:16	Reserved.
15:0	**NC and InitComplete**. Read-only. These bits provide duplicate versions of status bits F0x[F8, D8, B8, 98][NC and InitComplete] and F4x[F8, D8, B8, 98][NC and InitComplete] as follows:

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.

3.4 Function 1 Address Map Registers

See section 3.1 [Register Descriptions and Mnemonics] on page 137 for a description of the register naming convention. See section 2.11 [Configuration Space] on page 113 for details about how to access this space.

F1x00 Device/Vendor ID Register

Reset: 1201 1022h.

Bits	Description
31:16	**DeviceID: device ID**. Read-only.
15:0	**VendorID: vendor ID**. Read-only.

F1x08 Class Code/Revision ID Register

Reset: 0600 0000h.

Bits	Description
31:8	**ClassCode**. Read-only. Provides the host bridge class code as defined in the PCI specification.
7:0	**RevID: revision ID**. Read-only. Processor revision. 00h=A0.

F1x0C Header Type Register

Reset: 0080 0000h.

Bits	Description
31:0	**HeaderTypeReg**. Read-only. These bits are fixed at their default values. The header type field indicates that there are multiple functions present in this device.

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:

Base Address	Limit Address	Base Address	Limit Address
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 also section [The Northbridge Routing] 2.6.4.

DRAM mapping rules:

- Transaction addresses are within the defined range if:
  {DramBase[47:24], 00_0000h} <= address[47:0] <= {DramLimit[47:24], FF_FFFFh}.
- DRAM 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 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.
- See also section 2.6.4.1.1 [DRAM and MMIO Memory Space] on page 55.

Hoisting. When memory hoisting is enabled in a node (via F1xF0[DramHoleValid]), F1x[1, 0][7C:40][DramLimit] should be set up to account for the memory hoisted above the hole. I.e., F1x[1, 0][7C:40]DramLimit should be set to F1x[1, 0][7C:40][DramBase] plus the size of the amount of memory owned by the node plus the hole size (4G minus F1xF0[DramHoleBase]). See section 2.8.10 [Memory Hoisting] on page 105 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 section 2.8.9.2 [Node Interleaving] on page 104 for details.

F1x[78, 70, 68, 60, 58, 50, 48, 40] DRAM Base Address Registers

Bits	Description
31:16	**DramBase[39:24]: DRAM base address register bits[39:24]**. Read-write. Reset: X, except in F1x40 reset: 0000h.
15:11	Reserved

10:8	**IntlvEn[2:0]: interleave enable**. Read-write. Reset: X, except in F1x40 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 F1x124[DramIntlvEn].
7:2	Reserved.
1	**WE: write enable**. Read-write. Reset: 0. 1=Writes to this address range are enabled.
0	**RE: read enable**. Read-write. Reset: 0. 1=Reads to this address range are enabled.

F1x[178, 170, 168, 160, 158, 150, 148, 140] DRAM Base Address High Registers

Bits	Description
31:8	Reserved.
7:0	**DramBase[47:40]: DRAM base address register bits[47:40]**. Read-write. Reset: 0.

F1x[7C, 74, 6C, 64, 5C, 54, 4C, 44] DRAM Limit Address Registers

Bits	Description
31:16	**DramLimit[39:24]: DRAM limit address register bits[39:24]**. Read-write. Reset: X, except in F1x44 reset: FFFFh.
15:11	Reserved.
10:8	**IntlvSel: interleave select**. 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	**DstNode: destination Node ID**. 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	**DramLimit[47:40]: DRAM limit address register bits[47:40]**. 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 also section [The Northbridge Routing] 2.6.4.

MMIO mapping rules:

- Transaction addresses are within the defined range if:
  {00h, MMIOBase[39:16], 0000h} <= address[47:0] <= {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.
- Scenarios in which the address space of multiple MMIO ranges target the same IO device is supported.
- See also section 2.6.4.1.1 [DRAM and MMIO Memory Space] on page 55.

F1x[B8, B0, A8, A0, 98, 90, 88, 80] MMIO Base Address Registers

Bits	Description
31:8	**MMIOBase[39:16]: MMIO base address register bits[39:16]**. Read-write. Reset: X.
7:4	Reserved.
3	**Lock**. Read-write. Reset: X. 1=[The Memory Mapped IO Base/Limit Registers] F1x[BC:80], are read-only (including this bit). WE or RE in this register must be set in order for this to take effect.
2	**CpuDis: CPU Disable**. Read-write. Reset: X. 1=The MMIO range is disabled for CPU accesses; IO accesses still observe the MMIO range. If the MMIO range matches a DRAM range, this can be used to protect that DRAM space from IO devices by directing them to MMIO rather than DRAM.
1	**WE: write enable**. Read-write. Reset: 0. 1=Writes to this address range are enabled.
0	**RE: read enable**. Read-write. Reset: 0. 1=Reads to this address range are enabled.

F1x[BC, B4, AC, A4, 9C, 94, 8C, 84] MMIO Limit Address Registers

Bits	Description
31:8	**MMIOLimit[39:16]: MMIO limit address register bits[39:16]**. Read-write. Reset: X.
7	**NP: non-posted**. Read-write. Reset: X. 1=CPU write requests to this MMIO range are passed through the non-posted channel. This may be used to force writes to be non-posted for MMIO regions which map to the legacy ISA/LPC bus, or in conjunction with [The Link Transaction Control Register] F0x68[DsNpReqLmt] in order to allow downstream CPU requests to be counted and thereby limited to a specified number. This latter use of the NP bit may be used to avoid loop deadlock scenarios in systems that implement a region in an IO device that reflects downstream accesses back upstream. See the *HyperTransport™ IO Link Specification* summary of deadlock scenarios for more information. 0=CPU writes to this MMIO range use the posted channel. This bit does not affect requests that come from 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	**DstSubLink: destination sublink**. Read-write. Reset: X. When a link is unganged, 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 ganged, then this bit must be low.

5:4	**DstLink: destination link ID**. 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	**DstNode: destination node ID bits**. 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 also section [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 also section 2.6.4.1.2 [IO Space] on page 55.

F1x[D8, D0, C8, C0] IO-Space Base Address Registers

Bits	Description
31:25	Reserved.
24:12	**IOBase[24:12]: IO base address register bits[24:12]**. Read-write. Reset: X.
11:6	Reserved.
5	**IE: ISA enable**. Read-write. Reset: X. 1=The IO-space address window is limited to the first 256 bytes of each 1K byte block specified; this only applies to the first 64K bytes of IO space. 0=The PCI IO window is not limited in this way.
4	**VE: VGA enable**. Read-write. Reset: X. 1=Include IO-space transactions targeting the VGA-compatible address space within the IO-space window of this base/limit pair. These include IO accesses in which address bits[9:0] range from 3B0h to 3BBh or 3C0h to 3DFh (address bits[15:10] are not decoded); this only applies to the first 64K of IO space; i.e., address bits[24:16] must be low). 0=IO-space transactions targeting VGA-compatible address ranges are not added to the IO-space window. 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	**WE: write enable**. Read-write. Reset: 0. 1=Writes to this IO-space address range are enabled.
0	**RE: read enable**. Read-write. Reset: 0. 1=Reads to this IO-space address range are enabled.

F1x[DC, D4, CC, C4] IO-Space Limit Address Registers

Bits	Description
31:25	Reserved.
24:12	**IOLimit[24:12]: IO limit address register bits[24:12]**. Read-write. Reset: X.
11:7	Reserved.
6	**DstSubLink: destination sublink**. Read-write. Reset: X. When a link is unganged, this bit specifies the destination sublink of the link specified by F1x[DC:C0][DstLink]. 0=The destination link is sublink 0. 1=The destination link is sublink 1. If the link is ganged, then this bit must be low.
5:4	**DstLink: destination link ID**. 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	**DstNode: destination node ID bits**. 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 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 also section [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]} <= BusNo <= {BusNumLimit[7:0]}) & (DevCmpEn==0); or
  ({BusNumBase[4:0]} <= Device <= {BusNumLimit[4:0]}) & (DevCmpEn==1) & (BusNo == 00h).
- Configuration regions must not overlap each other.
- The appropriate RE or WE bit(s) must be set.
- See also section 2.6.4.1.3 [Configuration Space] on page 56.

Bits	Description
31:24	**BusNumLimit[7:0]: bus number limit bits[7:0]**. Read-write. Reset: X.
23:16	**BusNumBase[7:0]: bus number base bits[7:0]**. Read-write. Reset: X.
15:11	Reserved.
10	**DstSubLink: destination sublink**. Read-write. Reset: X. When a link is unganged, this bit specifies the destination sublink of the link specified by F1x[EC:E0][DstLink]. 0=The destination link is sublink 0. 1=The destination link is sublink 1. If the link is ganged, then this bit must be low.
9:8	**DstLink: destination link ID**. 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	**DstNode: destination node ID bits**. Read-write. Reset: X. For transactions within the this configuration-space range, this field specifies the destination node ID.
3	Reserved.
2	**DevCmpEn: device number compare mode enable**. 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	**WE: write enable**. Read-write. Reset: 0. 1=Writes to this configuration-space address range are enabled.
0	**RE: read enable**. 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	**DramHoleBase[31:24]**. 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, DramHoleBase >= MSRC001_001A[TOM[31:24]]. See section 2.8.10 [Memory Hoisting] on page 105 for additional programming information.
23:16	Reserved.
15:7	**DramHoleOffset[31:23]: DRAM hole offset address**. Read-write. When memory hosting is enabled, this value is subtracted from the physical address of certain transactions before being passed to the DCT. See section 2.8.10 [Memory Hoisting] on page 105 for additional programming information.
6:2	Reserved.
1	**DramMemHoistValid**. 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 section 2.8.10 [Memory Hoisting] on page 105 for additional programming information.
0	**DramHoleValid**. 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 section 2.8.10 [Memory Hoisting] on page 105 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	**DstSubLink: destination sublink**. Read-write. When a link is unganged, 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 ganged, then this bit must be low.

13:12	**DstLink: destination link ID**. 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	**DstNode[2:0]: destination node ID**. Read-write. For transactions within the F1xF4[VE]-defined range, this field specifies the destination node ID.
3	**Lock**. Read-write. Reset: 0. 1=All the bits in this register (F1xF4) are read-only (including this bit).
2	**CpuDis: CPU Disable**. Read-write. 1=The F1xF4[VE]-defined MMIO range is disabled for CPU accesses; i.e., CPU accesses to this range are treated as if the VE=0.
1	**NP: non-posted**. Read-write. 1=CPU write requests to the F1xF4[VE]-defined MMIO range are passed through the non-posted channel. 0=CPU writes may be posted.
0	**VE: VGA enable**. Read-write. 1=Transactions targeting the VGA-compatible address space are routed and controlled as specified by this register. 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]=1b.

Before reading F1x114_x2 or F1x114_x3 software must initialize the registers or NB Array MCA errors may occur. BIOS should initialize index 0h of F1x114_x2 and F1x114_x3 to prevent reads from F1x114 from generating NB Array MCA errors.

.

Bits	Description
31	Reserved.
30:28	**AddrMapType**. 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	**Index**. 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[44:23] | 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] on page 55.

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:

Write base bit	Write mask bit	Read base bit	Read mask bit	Notes
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	**MmioMapBase[20:0]**. Read-write. Reset: X.
7	Reserved.
6	**MmioDstThisNode**. Read-write. Reset: X. 1=The destination is the local node. See MmioDstNode.
5:3	Reserved.
2:0	**MmioDstNode**. 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 unganged).

F1x114_x3 Extended MMIO Address Mask Registers

See F1x114_x2 for details.

Bits	Description
31:29	Reserved.
28:8	**MmioMapMask[20:0]**. Read-write. Reset: X. 1=Address bit is a don't care.
7:1	Reserved.
0	**MmioMapEn**. 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 section 2.8.9.2 [Node Interleaving] on page 104 for more details.

Bits	Description
31:24	Reserved.
23:21	**DramIntlvSel: interleave select**. 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	**DramBaseAddr[47:27]: dram base address**. Read-write. Reset: 0.

F1x124 DRAM Limit System Address Register

See F1x120.

Bits	Description
31:21	Reserved.
23:21	**DramIntlvEn[2:0]: dram interleave enable**. 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 F1x[1, 0][7C:40][IntlvEn].
20:0	**DramLimitAddr[47:27]: dram limit address**. Read-write. Reset: 1F_FFFFh.

3.5 Function 2 DRAM Controller Registers

See section 3.1 [Register Descriptions and Mnemonics] on page 137 for a description of the register naming convention. See section 2.11 [Configuration Space] on page 113 for details about how to access this space.

F2x00 Device/Vendor ID Register

Reset: 1202 1022h.

Bits	Description
31:16	**DeviceID: device ID**. Read-only.
15:0	**VendorID: vendor ID**. Read-only.

F2x08 Class Code/Revision ID Register

Reset: 0600 0000h.

Bits	Description
31:8	**ClassCode**. Read-only. Provides the host bridge class code as defined in the PCI specification.
7:0	**RevID: revision ID**. Read-only.

F2x0C Header Type Register

Reset: 0080 0000h.

Bits	Description
31:0	**HeaderTypeReg**. Read-only. These bits are fixed at their default values. The header type field indicates that there are multiple functions present in this device.

F2x[1, 0][5C:40] DRAM CS Base Address Registers

Reset: 0000 0000h. See section 2.8.1 [DCT Configuration Registers] on page 65 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 section 2.8 [DRAM Controllers (DCTs)] on page 64.

The processor logically supports the following number of DIMMs in the following packages:

Table 37: DIMM support per package

Package	Number of DIMMs per channel			
	Registered	4-Rank registered	Unbuffered	SO-DIMMs
Fr2(1207)	4	2	0	0
G(1207)	4	2	0	0
AM2r2 and AM3	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 38: Logical DIMM, Chip Select, CKE, ODT, and Register Mapping

Base Address Registers	Mask Register	Logical DIMM[1]		Chip Select	M[B, A]_ CKE[x]	ODT
		N	R4			
F2x[1, 0]40	F2x[1, 0]60	0	0	M[B, A]0_CS_L[0]	0	M[B, A]0_ODT[0]
F2x[1, 0]44				M[B, A]0_CS_L[1]	1	M[B, A]0_ODT[0][3] M[B, A]1_ODT[0][4]
F2x[1, 0]48	F2x[1, 0]64	1	1	M[B, A]1_CS_L[0][3]	0	M[B, A]1_ODT[0][3] M[B, A]0_ODT[0][4]
F2x[1, 0]4C				M[B, A]1_CS_L[1][3]	1	M[B, A]1_ODT[0][3]
F2x[1, 0]50	F2x[1, 0]68	2[2]	0	M[B, A]2_CS_L[0]	0	M[B, A]2_ODT[0]
F2x[1, 0]54				M[B, A]2_CS_L[1]	1	M[B, A]2_ODT[0]
F2x[1, 0]58	F2x[1, 0]6C	3[2]	1	M[B, A]3_CS_L[0]	0	M[B, A]3_ODT[0]
F2x[1, 0]5C				M[B, A]3_CS_L[1]	1	M[B, A]3_ODT[0]

Notes:
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 the AM2r2 and AM3 packages.
3. Fr2(1207), AM2r2, and AM3 packages.
4. SO-DIMM using an AM2r2 or AM3 package.

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 also section 2.6.1 [Northbridge (NB) Architecture] on page 54.

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 section 2.8.9.1 [Chip Select Interleaving] on page 102 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	**BaseAddr[36:27]: normalized physical base address bits [36:27]**. Read-write.
18:14	Reserved.
13:5	**BaseAddr[21:13]: normalized physical base address bits [21:13]**. Read-write.
4	Reserved.
3	**OnDimmMirror: on-DIMM mirroring (ODM) enabled**. Read-write. 1=Address and bank bits are swapped for this chip select in order to account for swapped routing on the DIMM during DRAM initialization. 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: • M[B, A]_BANK[0] and M[B, A]_BANK[1]. • M[B, A]_ADD[3] and M[B, A]_ADD[4]. • M[B, A]_ADD[5] and M[B, A]_ADD[6]. • M[B, A]_ADD[7] and M[B, A]_ADD[8].
2	**TestFail: memory test failed**. Read-write. This bit is set by BIOS to indicate that the rank is present but the memory is bad. The CSEnable and Spare bits must not be set if this bit is set. This bit must be set prior to DRAM initialization.
1	**Spare: spare rank**. Read-write. This bit identifies the chip select associated with the spare rank. See section 2.8.11 [On-Line Spare] on page 107.
0	**CSEnable: chip select enable**. Read-write.

F2x[1, 0][6C:60] DRAM CS Mask Registers

Reset: 0000 0000h. See section 2.8.1 [DCT Configuration Registers] on page 65 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	**AddrMask[36:27]: normalized physical address mask bits [36:27]**. Read-write.
18:14	Reserved.
13:5	**AddrMask[21:13]: normalized physical address mask bits [21:13]**. Read-write.
4:0	Reserved.

F2x[1, 0]78 DRAM Control Register

Reset: 0000 0006h. See section 2.8.1 [DCT Configuration Registers] on page 65 for general programming information about DCT configuration registers.

Bits	Description
31:22	**MaxRdLatency: maximum read latency**. 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 DRAM controller uses this to help determine when incoming DRAM read data can be safely transferred to the NCLK domain. The time is specified in NB clocks and includes the asynchronous and synchronous latencies. See section 2.8.8.8.4 [Calculating MaxRdLatency] on page 98 for information on how to program this field.
21:20	Reserved.
19	**EarlyArbEn: early arbitration enable**. 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.
18	**DqsRcvEnTrain: DQS receiver enable training mode**. Read-write. 1=Enable DQS receiver enable training mode. 0 = Normal DQS receiver enable operation.
17:16	Reserved.
15	**ChSetupSync: channel setup synchronize**. 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 F2x[1, 0]9C_x04. 0=The DRAM controller derives its setup information from F2x[1, 0]9C_x04. Note: BIOS must set this bit to synchronize setup information between the two channels when both of the following conditions are true: • The DCTs are in ganged mode. • F2x[1, 0]9C_x04[AddrCmdSetup, CsOdtSetup, CkeSetup] setups for one DCT are all 0s and at least one of the setups, F2x[1, 0]9C_x04[AddrCmdSetup, CsOdtSetup, CkeSetup], of the other controller is 1.
15:14	Reserved.
13:12	**Trdrd[3:2]: read to read timing**. Read-write. This field along with F2x[1, 0]8C[Trdrd[1:0]] combine to specify a 4-bit value, Trdrd[3:0], when F2x[1, 0]94[Ddr3Mode]=1. See F2x[1, 0]8C[Trdrd[1:0]].
11:10	**Twrwr[3:2]: write to write timing**. Read-write. This field along with F2x[1, 0]8C[Twrwr[1:0]] combine to specify a 4-bit value, Twrwr[3:0], when F2x[1, 0]94[Ddr3Mode]=1. See F2x[1, 0]8C[Twrwr[1:0]].
9:8	**Twrrd[3:2]: write to read DIMM termination turnaround**. Read-write. This field along with F2x[1, 0]8C[Twrrd[1:0]] combine to specify a 4-bit value, Twrrd[3:0], when F2x[1, 0]94[Ddr3Mode]=1. See F2x[1, 0]8C[Twrrd[1:0]].

7:4	Reserved.
3:0	**RdPtrInit: read pointer initial value**. 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. It is recommended that these bits remain in the default state. 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 — Reserved 0111b - 1111b — Reserved

F2x[1, 0]7C DRAM Initialization Register

Reset: 0000 0000h. See section 2.8.1 [DCT Configuration Registers] on page 65 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	**EnDramInit: enable DRAM initialization**. Read-write. 1=Place the DRAM controller in the BIOS-controlled DRAM initialization mode. The DCT asserts memory reset and deasserts CKE when this bit is set. BIOS must wait until F2x[1, 0]98[DctAccessDone] = 1 before programming AssertCke=1 and DeassertMemRstX=1. BIOS must clear this bit after DRAM initialization is complete. See also sections 2.8.8.5.1 [Software DDR2 Device Initialization] on page 80 and 2.8.8.5.2 [Software DDR3 Device Initialization] on page 82.
30	**SendControlWord: send control word**. Read; write-1-only. 1= The DCT sends a control word to a chip select pair defined in F2x[1, 0]A8[CtrlWordCS]. This bit is cleared by hardware after the command completes. This bit is valid only when F2x[1, 0]94[Ddr3Mode] = 1 and F2x[1, 0]90[UnbufferedDimm] = 0.
29	**SendZQCmd: send ZQ command**. Read; write-1-only. 1=The DCT sends the ZQ calibration command. This bit is cleared by the hardware after the command completes. This bit is valid only when F2x[1, 0]94[Ddr3Mode] = 1.
28	**AssertCke: assert CKE**. Read-write. Setting this bit causes the DCT to assert the CKE pins. This bit cannot be used to deassert the CKE pins.
27	**DeassertMemRstX: deassert memory reset**. Read-write. Setting this bit causes the DCT to deassert the memory reset . This bit cannot be used to assert the memory reset pin.
26	**SendMrsCmd: send MRS/EMRS command**. Read; write-1-only. 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.

25	**SendAutoRefresh: send auto refresh command**. Read; write-1-only. 1=The DCT sends an auto refresh command. This bit is cleared by hardware after the command completes.
24	**SendPchgAll: send precharge all command**. 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	**MrsChipSel: MRS/EMRS command chip select**. 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. 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	**MrsBank: bank address for MRS/EMRS commands**. Read-write. This field specifies the data driven on the DRAM bank pins for MRS and EMRS commands.
15:0	**MrsAddress: address for MRS/EMRS commands**. Read-write. This field specifies the data driven on the DRAM address pins 15-0 for MRS and EMRS commands.

F2x[1, 0]80 DRAM Bank Address Mapping Register

Reset: 0000 0000h. See section 2.8.1 [DCT Configuration Registers] on page 65 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 39, for DDR2, Table 40, 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	**Dimm3AddrMap: DIMM 3 address map**. Read-write.
11:8	**Dimm2AddrMap: DIMM 2 address map**. Read-write.
7:4	**Dimm1AddrMap: DIMM 1 address map**. Read-write.
3:0	**Dimm0AddrMap: DIMM 0 address map**. Read-write.

Table 39: 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	x	14	13	Row	x	x	x	17	16	15	27	26	25	24	23	22	21	20	19	18
		512Mb, x16				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	15	14	13	Row	x	x	17	16	29	28	27	26	25	24	23	22	21	20	19	18
		2G, x16				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	15	14	13	Row	x	17	16	30	29	28	27	26	25	24	23	22	21	20	19	18
		4Gb, x16				Col	x	x	x	x	x	AP	12	11	10	9	8	7	6	5	4	3
1000b	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 40: DDR3 DRAM address mapping

Bits	CS Size	Device size, width	Bank			Address																
			2	1	0		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0000b		Reserved				Row																
						Col																
0001b	256MB	512Mb, x16	15	14	13	Row	x	x	x	x	17	16	27	26	25	24	23	22	21	20	19	18
						Col	x	x	x	x	x	AP	12	11	10	9	8	7	6	5	4	3
0010b	512MB	512Mb, x8	15	14	13	Row	x	x	x	17	16	28	27	26	25	24	23	22	21	20	19	18
		1Gb, x16				Col	x	x	x	x	x	AP	12	11	10	9	8	7	6	5	4	3
0011b		Reserved				Row																
						Col																
0100b		Reserved				Row																
						Col																
0101b	1GB	1Gb, x8	15	14	13	Row	x	x	17	16	29	28	27	26	25	24	23	22	21	20	19	18
		2Gb, x16				Col	x	x	x	x	x	AP	12	11	10	9	8	7	6	5	4	3
0110b	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

Table 40: 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
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	15	14	13	Row	17	16	31	30	29	28	27	26	25	24	23	22	21	20	19	18
		8Gb, x16				Col	x	x	x	x	x	AP	12	11	10	9	8	7	6	5	4	3
1011b	8GB	4Gb, x4	16	15	14	Row	17	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18
		8Gb, x8				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 section 2.8.1 [DCT Configuration Registers] on page 65 for general programming information about DCT configuration registers.

Bits	Description
31:27	Reserved.
26	**MprEn: multi purpose register enable**. 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	**MprLoc: multi purpose register address location**. 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	**PchgPDModeSel: precharge power down mode select**. Read-write. 0=DDR3-defined slow exit mode; 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. 1=DDR3-defined fast exit mode; the DCT issues the first valid command a minimum of 7.5ns for DDR3-800 and DDR3-1066 DRAMs or 6ns for DDR3-1333 and DDR3-1600 DRAMs (DDR3-defined tXP) after precharge power down exit. This bit is valid only when F2x[1, 0]94[Ddr3Mode] = 1. BIOS should set this bit when there is one 2 rank SO-DIMM on a populated channel; otherwise, this bit is always 0.
22:20	**Tcwl: CAS write latency**. Read-write. This specifies the number of clock cycles from internal write command to first write data in at the DRAM. 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	**SRT: self refresh temperature range**. 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.

<table>
<tr><td>18</td><td>ASR: auto self refresh. 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.</td></tr>
<tr><td>17:14</td><td>Reserved.</td></tr>
<tr><td>13</td><td>Qoff: output disable. Read-write. Specifies the QOFF value for the DRAM devices. 0=Output buffers enabled. 1=Output buffers disabled.</td></tr>
<tr><td>12</td><td>Reserved.</td></tr>
<tr><td>11:10</td><td>DramTermDyn: DRAM dynamic termination. 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 (F2x[1, 0]90[InitDram]). BIOS must enable dynamic termination when there are 2 DIMMs on a channel. This field is valid only when F2x[1, 0]94[Ddr3Mode] = 1.
<table>
<tr><th>Bits</th><th>DDR3 definition</th></tr>
<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>
</table></td></tr>
<tr><td>9:7</td><td>DramTerm: DRAM nominal termination. 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 (F2x[1, 0]90[InitDram]). This field is valid only when F2x[1, 0]94[Ddr3Mode] = 1. For DDR2 termination values, see F2x[1, 0]90[DramTerm].
<table>
<tr><th>Bits</th><th>DDR3 definition</th><th>Bits</th><th>DDR3 definition</th></tr>
<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>
</table></td></tr>
<tr><td>6:4</td><td>Twr: write recovery. Read-write. This specifies the minimum time from the last data write until the chip-select bank precharge; this is the WR field in the DDR3 specification. See F2x[1, 0]88[Twr] for DDR2 values.This field specifies the encodings for DDR2-1066 when F2x[1, 0]94[Ddr3Mode] = 0 and when F2x[1, 0]94[MemClkFreq]=100b.
<table>
<tr><th>Bits</th><th>DDR2-1066 definition</th><th>Bits</th><th>DDR3 definition</th></tr>
<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>
</table></td></tr>
</table>

3:2	**DrvImpCtrl: drive impedance control**. Read-write. This field specifies impedance of the DRAM output driver. This field is valid only when F2x[1, 0]94[Ddr3Mode] = 1. Bits DDR3 definition 00b 40 ohm driver; Ron40 = Rzq/6 (40 ohm with nominal Rzq=240 ohms) 01b 34 ohm driver; Ron34 = Rzq/7 (34 ohm with nominal Rzq=240 ohms) 10b Reserved for 30 ohm driver; Ron30 = Rzq/8 (30 ohm with nominal Rzq=240 ohms) 11b Reserved
1:0	**BurstCtrl: burst length control**. Read-write. Specifies the number of sequential beats of DQ related to one read or write command. This field interacts with F2x[1, 0]90[Width128] as follows: • If Width128 = 0, then BurstCtrl is programmed to 00b (8-beat burst length; 64-byte access). • If Width128 = 1, then BurstCtrl is programmed to 10b (4-beat burst length; 64-byte access). All other encodings are reserved.

F2x[1, 0]88 DRAM Timing Low Register

Reset: FF00 0000h. See section 2.8.1 [DCT Configuration Registers] on page 65 for general programming information about DCT configuration registers.

Bits	Description
31:24	**MemClkDis: MEMCLK disable**. Read-write. 1=Disable the MEMCLK. The bits MemClkDis[7:0] are mapped to packages as follows: Bit Fr2(1207) pin name AM2r2 pin name AM3 pin name 0 N/A M[B,A]1_CLK_H/L[1] M[B,A]_CLK_H/L[0] 1 N/A M[B,A]0_CLK_H/L[1] M[B,A]_CLK_H/L[1] 2 M[B,A]3_CLK N/A M[B,A]_CLK_H/L[2] 3 M[B,A]2_CLK N/A M[B,A]_CLK_H/L[3] 4 M[B,A]1_CLK M[B,A]1_CLK_H/L[0] M[B,A]_CLK_H/L[4] 5 M[B,A]0_CLK M[B,A]0_CLK_H/L[0] M[B,A]_CLK_H/L[5] 6 N/A M[B,A]1_CLK_H/L[2] M[B,A]_CLK_H/L[6] 7 N/A M[B,A]0_CLK_H/L[2] M[B,A]_CLK_H/L[7] Note: F2x88 controls the channel A memory clock pins and F2x188 controls the channel B memory clock pins.
23:22	**Trrd: row to row delay (or RAS to RAS delay)**. 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. Bits DDR2 definition Bits DDR2-1066/DDR3 definition 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 **Twr: write recovery time**. 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].

Bits	DDR2 definition
00b	3 clocks
01b	4 clocks
10b	5 clocks
11b	6 clocks

19:16 DDR2
20:16 DDR2 1066/ DDR3 **Trc: row cycle time**. 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.

Bits	DDR2 definition	Bits	DDR2-1066/DDR3 definition
0h	11 clocks	00h	11 clocks
1h	12 clocks	01h	12 clocks
...	...	...	...
Fh	26 clocks	1Fh	42 clocks

15:12 **Tras: row active strobe**. 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.

Bits	DDR2 definition	Bits	DDR2-1066/DDR3 definition
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 **Trtp: read to precharge time**. 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, F2x[1, 0]94[Ddr3Mode]. 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.

Bits	DDR2 definition	Recommended value.
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, DDR1066)
	5 clocks for burst length of 64 bytes	(DDR667, DDR800, DDR1066)

Bits	DDR3 definition	Recommended value.
00b	4 clocks	(DDR800, DDR1066)
01b	5 clocks	(DDR1333)
10b	6 clocks	
11b	7 clocks	

9:7	**Trp: row precharge time**. 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 (F2x[1, 0]94[Ddr3Mode]) and when F2x[1, 0]94[MemClk-Freq]=100b to support DDR2-1066. Bits DDR2 definition — Bits DDR2-1066/DDR3 definition 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	**Trcd: RAS to CAS delay**. 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. Bits DDR2 definition — Bits DDR2-1066/DDR3 definition 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	**Tcl: CAS latency**. Read-write. This specifies the time from the CAS assertion for a read cycle until data return (from the perspective of the DRAM devices). For DDR2, write CAS latency is always read CAS latency minus 1. The DCT adjusts these latencies appropriately for registered DIMMs. The definition of this field varies with the DDR type (F2x[1, 0]94[Ddr3Mode]). Bits DDR2 definition — Bits DDR3 definition 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 3. 0111b - 1111bReserved

F2x[1, 0]8C DRAM Timing High Register

Reset: 0000 0000h. See section 2.8.1 [DCT Configuration Registers] on page 65 for general programming information about DCT configuration registers.

Bits	Description
31:29	**Trfc3: auto-refresh row cycle time for logical DIMM 3**. Read-write. See Trfc0.
28:26	**Trfc2: auto-refresh row cycle time for logical DIMM 2**. Read-write. See Trfc0.
25:23	**Trfc1: auto-refresh row cycle time for logical DIMM 1**. Read-write. See Trfc0.

<table>
<tr><td>22:20</td><td>Trfc0: auto-refresh row cycle time for logical DIMM 0. Read-write. This specifies the minimum time from an auto-refresh command to an activate command or another auto refresh command. 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].
<table>
<tr><th>Bits</th><th>DDR2 definition</th><th>DDR3 definition</th></tr>
<tr><td>000b</td><td>75 ns (all speeds, 256 Mbit)</td><td>Reserved</td></tr>
<tr><td>001b</td><td>105 ns (all speeds, 512 Mbit)</td><td>90 ns (all speeds, 512 Mbit)</td></tr>
<tr><td>010b</td><td>127.5 ns (all speeds, 1 Gbit)</td><td>110 ns (all speeds, 1 Gbit)</td></tr>
<tr><td>011b</td><td>195 ns (all speeds, 2 Gbit)</td><td>160 ns (all speeds, 2 Gbit)</td></tr>
<tr><td>100b</td><td>327.5 ns (all speeds, 4 Gbit)</td><td>300 ns (all speeds, 4 Gbit)</td></tr>
<tr><td>101b</td><td>Reserved</td><td>350 ns (all speeds, 8 Gbit)</td></tr>
<tr><td>110b-111b</td><td>Reserved</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>19</td><td>Reserved.</td></tr>
<tr><td>18</td><td>DisAutoRefresh: disable automatic refresh. Read-write. 1=Automatic refresh is disabled. This is sometimes useful during electrical characterization.</td></tr>
<tr><td>17:16</td><td>Tref: refresh rate. Read-write. This specifies the average time between refresh requests to all DRAM devices.
<table>
<tr><th>Bits</th><th>Definition</th></tr>
<tr><td>00b</td><td>Undefined behavior.</td></tr>
<tr><td>01b</td><td>Reserved</td></tr>
<tr><td>10b</td><td>Every 7.8 microseconds</td></tr>
<tr><td>11b</td><td>Every 3.9 microseconds</td></tr>
</table></td></tr>
<tr><td>15:14</td><td>Trdrd[1:0]: read to read timing. Read-write. 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. 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 F2x[1, 0]78[Trdrd[3:2]] combine to specify a 4-bit value, Trdrd[3:0], when F2x[1, 0]94[Ddr3Mode]=1.
<table>
<tr><th>Bits</th><th>DDR2 definition</th><th>Bits</th><th>DDR3 definition</th></tr>
<tr><td>00b</td><td>2 clocks</td><td>0000b</td><td>2 clocks</td></tr>
<tr><td>01b</td><td>3 clocks</td><td>0001b</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>
</table>
See section [The Trdrd (Read to Read Timing)] 2.8.8.4.1 for information on how to program this field.</td></tr>
</table>

13:12 **Twrwr[1:0]: write to write timing**. Read-write. Twrwr specifies the minimum number of cycles from the last clock of virtual CAS of the first write-burst operation to the clock in which CAS is asserted for a following write-burst operation that changes the enabled terminator. 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 F2x[1, 0]78[Twrwr[3:2]] combine to specify a 4-bit value, Twrwr[3:0], when F2x[1, 0]94[Ddr3Mode]=1.

Bits	DDR2 definition	Bits	DDR3 definition
00b	1 clock	0000b - 0001b	Reserved
01b	2 clocks	0010b	3 clocks
10b	3 clocks	...	...
11b	4 clocks	1001b	10 clocks
		1010b - 1111b	Reserved

See section [The Twrwr (Write to Write Timing)] 2.8.8.4.2 for information on how to program this field.

11:10 **Twrrd[1:0]: write to read DIMM termination turnaround**. 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.[1] 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.[2] This field along with F2x[1, 0]78[Twrrd[3:2]] combine to specify a 4-bit value, Twrrd[3:0], when F2x[1, 0]94[Ddr3Mode]=1.

Bits	DDR2 definition	Bits	DDR3 definition
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

Notes:

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 Twrrd.

See section [The Twrrd (Write to Read DIMM Termination Turn-around)] 2.8.8.4.3 for information on how to program this field.

9:8 **Twtr: internal DRAM write to read command delay**. 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, F2x[1, 0]94[Ddr3Mode] and when F2x[1, 0]94[MemClkFreq]=100b to support DDR2-1066.

Bits	DDR2 definition	Bits	DDR2-1066 definition/DDR3 definition
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	**TrwtTO: read to write turnaround for data, DQS contention**. 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. Bits DDR2 definition / Bits DDR3 definition 0000b Reserved / 0000b Reserved 0001b 3 clocks / 0001b 3 clocks 0010b 4 clocks / 0010b 4 clocks ... ... / ... ... 0111b 9 clocks / 1111b 17 clocks 1000b - 1111b Reserved See section [The TrwtTO (Read-to-Write Turnaround for Data, DQS Contention)] 2.8.8.4.4 for information on how to program this field.
3:0	**TrwtWB: read to write turnaround for opportunistic write bursting**. 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. Bits DDR2 definition / Bits DDR3 definition 0000b 3 clocks / 0000b 3 clocks 0001b 4 clocks / 0001b 4 clocks ... ... / ... ... 0111b 10 clocks / 1111b 18 clocks 1000b - 1111b Reserved See section [The TrwtWB (Read-to-Write Turnaround for Opportunistic Write Bursting)] 2.8.8.4.5 for information on how to program this field.

F2x[1, 0]90 DRAM Configuration Low Register

Reset: 0000 0000h, except bit 16 (see below). See section 2.8.1 [DCT Configuration Registers] on page 65 for general programming information about DCT configuration registers.

Bits	Description
31:24	Reserved.
23	**ForceAutoPchg: force auto precharging**. Read-write. 1=Force auto-precharge cycles with every read or write command. This may be preferred in situations where power savings is favored over performance.
22:21	**IdleCycLowLimit: idle cycle low limit**. 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 F2x[1, 0]90[DynPageCloseEn] = 0. Bits Definition 00b 16 clocks 01b 32 clocks 10b 64 clocks 11b 96 clocks
20	**DynPageCloseEn: dynamic page close enable**. Read-write. 1=The DRAM controller dynamically determines when to close open pages based on the history of that particular page and F2x[1, 0]90[IdleCycLowLimit]. 0=Any open pages not auto-precharged by the DRAM controller are automatically closed after 128 clocks of inactivity.

19	**DimmEccEn: DIMM ECC enable**. Read-write. 1=ECC checking is capable of being enabled for all DIMMs on the DRAM controller (through F3x44[DramEccEn]). This bit should not be set unless all populated DIMMs support ECC check bits. 0=ECC checking is disabled on the DRAM controller.
18	**PendRefPayback: pending refresh payback**. 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.
17	**EnterSelfRef: enter self refresh command**. 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 section 2.8.8.8 [DRAM Training] on page 86 for information on how to use this bit.
16	**UnbuffDimm: unbuffered DIMM**. 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	**X4Dimm: x4 (by 4) DIMMs**. 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	**Width128: width of DRAM interface in 128-bit mode**. Read-write. 1=The DRAM controller interface is 2 DIMMs wide. 0=The DRAM controller interface is 1 DIMM wide.
10	**BurstLength32: DRAM burst length set for 32 bytes**. 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: 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 is undefined when F2x[1, 0]94[Ddr3Mode]=1.
9	**SelfRefRateEn: faster self refresh rate enable**. 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	**ParEn: parity enable**. Read-write. 1=Enables address parity computation output, PAR, and enables the parity error input, ERR. This bit is valid only when UnbuffDimm=0.
7	**DramDrvWeak: DRAM drivers weak mode**. 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	**DisDqsBar: disable low differential DQS pin**. 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	**DramTerm: DRAM termination**. 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]. Bits DDR2 definition 00b On die termination disabled. 01b 75 ohms. 10b 150 ohms. 11b 50 ohms.
3:2	**PllLockTime: registered DIMM PLL lock time**. Read-write. This specifies registered DIMM PLL lock time as follows when F2x[1, 0]94[Ddr3Mode]=0: Bits Definition 00b 15us 01b 6us 1xb Reserved
1	**ExitSelfRef: exit self refresh (after suspend to RAM or for DRAM training) command**. 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. See section 2.8.8.8.2 [DDR2 DRAM Training] on page 91. 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.
0	**InitDram: initialize DRAM**. 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. See section 2.8.8.5 [DRAM Device Initialization] on page 79 for more details.

F2x[1, 0]94 DRAM Configuration High Register

Reset: 0008 0200h. See section 2.8.1 [DCT Configuration Registers] on page 65 for general programming information about DCT configuration registers.

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>31:28</td><td>FourActWindow[3:0]: four bank activate window. 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:
<table>
<tr><th>Bits</th><th>DDR2 window size</th><th>Bits</th><th>DDR2-1066/DDR3 window size</th></tr>
<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>
</table>
See section 2.8.8.4.6 [FourActWindow (Four Bank Activate Window or tFAW)] on page 73 for information on how to program this field.</td></tr>
<tr><td>27:24</td><td>DcqBypassMax: DRAM controller queue bypass maximum. 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.
<table>
<tr><th>Bits</th><th>Definition</th></tr>
<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>
</table></td></tr>
<tr><td>23</td><td>ProcOdtDis: processor on-die termination disable. Read-write. 1=The processor-side on-die termination is disabled. 0=Processor-side on-die termination enabled. See F2x[1, 0]9C_x00[ProcOdt] for ODT definitions.</td></tr>
<tr><td>22</td><td>BankSwizzleMode: bank swizzle mode. 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 39 of F2x[1, 0]80, are remapped as follows:

Define X as a bank address bit (e.g., X=15 if the bank bit is specified to be address bit 15).
X' = X, if the DCT is in 64-bit unganged mode, or X+1 in 128-bit ganged mode.
Define S(n) as the state of address bit n (0 or 1) and B as the remapped bank address bit. Then,
B= S(X') ^ S(X' + 2) ^ S(X' + 4); for a 4-bank DRAM.
B= S(X') ^ S(X' + 3) ^ S(X' + 6); for an 8-bank DRAM.

For example, encoding 02h of Table 39 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}.

BIOS should set this bit to 1b.</td></tr>
<tr><td>21</td><td>FreqChgInProg: frequency change in progress. 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.</td></tr>
</table>

20	**SlowAccessMode: slow access mode (a.k.a. 2T mode)**. 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	**DcqArbBypassEn: DRAM controller arbiter bypass enable**. Read-write. 0=DCQ entries are always passed through the arbiter. 1=Bypass the arbitration logic when there is only one entry in the DRAM controller queue entry.
18	**FourRankRDimm: four rank registered DIMM connected**. Read-write. 1=Four-rank registered DIMMs are connected to the channel. In this mode, only two DIMMs per channel are supported. See F2x[1, 0][5C:40] for configuration information in this mode.
17	Reserved.
16	**PowerDownMode: power down mode**. Read-write. This specifies how a DIMM or group of DIMMs enters power down mode, when enabled by F2x[1, 0]94[PowerDownEn]. A DIMM enters power down mode when the DCT deasserts the CKE pin to that DIMM. The command and address signals tristate one MEMCLK after CKE deasserts. There are two CKE pins per DRAM channel. For each channel: Bit Description 0b Channel CKE control mode. The DRAM channel is placed in power down mode when all chip selects associated with the channel are idle. Both CKE pins for the channel operate in lock step, in terms of placing the channel DIMMs in power down mode. - CKE0 is connected to CS0, CS1, CS4, CS5 (the even DIMMs); - CKE1 is connected to CS2, CS3, CS6, CS7 (the odd DIMMs). 1b Reserved.
15	**PowerDownEn: power down mode enable**. 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.
14	**DisDramInterface: disable the DRAM interface**. Read-write. 1=The DRAM controller is disabled and the DRAM interface is placed into a low power state. This bit must be set if there are no DIMMs connected to the DCT.
13	**DisSimulRdWr: disable simultaneous read and write**. 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 (F2x110[DctGangEn]). The value of this bit should be programmed the same for both DCTs.
12	**RDqsEn: read DQS enable**. Read-write. This is applied to the DRAM device's DDR2-defined EMRS(1) or DDR3-defined MR1 registers during DRAM initialization (see section 2.8.8.5 [DRAM Device Initialization] on page 79). 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 F2x[1, 0]94[Ddr3Mode] as follows: DDR2 definition: 1=The DIMM DM pins are used as read DQS pins and data masking is disabled. 0=DM pins function as data mask pins. DDR3 definition: 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	**ZqcsInterval: ZQ calibration short interval**. 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 F2x[1, 0]94[Ddr3Mode] = 1. Bits Definition 00b ZQ calibration short command is disabled 01b 64 ms 10b 128 ms 11b 256 ms
9	**LegacyBiosMode**. Read-write. 0=Normal DCT functionality. 1=Legacy BIOS mode is enabled and DCT1 is disabled. In this mode, [The DRAM DQS Receiver Enable Timing Control Registers] F2x[1, 0]9C_x[2B:10] value in DqsRcvEnGrossDelay[0] is programmed into the register fields of DqsRcvEnGrossDelay[1, 2, 3, 4, 5, 6, 7] and DqsRcvEnGrossDelayCheck for each corresponding DIMM (as defined by [The DRAM CS Base Address Registers] F2x[1, 0][5C:40]). The value in the field DqsRcvEnFineDelay[0] is programmed into the register fields of DqsRcvEnFineDelay[1, 2, 3, 4, 5, 6, 7] and DqsRcvEnFineDelayCheck. While this bit is set, MemClkFreq is fixed at 200 MHz and accesses to registers F2x[1, 0]9C_x[302:301, 202:201, 102:101, 02:01], F2x[1, 0]9C_x[303, 203, 103, 03], F2x[1, 0]9C_x[306:305, 206:205, 106:105, 06:05], F2x[1, 0]9C_x[307, 207, 107, 07], and F2x[1, 0]9C_x[45:30] 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 F2x[1, 0]9C_x[2B:10].
8	**Ddr3Mode**. Read-write. 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	**MemClkFreqVal: memory clock frequency valid**. Read-write. System BIOS should set this bit after setting up F2x[1, 0]94[MemClkFreq] to the proper value. This indicates to the DRAM controller that it may start driving MEMCLK at the proper frequency. BIOS should poll FreqChgInProg 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 section 2.8.8.5 [DRAM Device Initialization] on page 79.
2:0	**MemClkFreq: memory clock frequency**. Read-write. This field specifies the frequency of the DRAM interface (MEMCLK). The definiton varies with the DDR type, F2x[1, 0]94[Ddr3Mode]. Bits DDR2 Definition DDR3 Definition 000b 200 MHz Reserved 001b 266 MHz Reserved 010b 333 MHz Reserved 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 section 2.8.1 [DCT Configuration Registers] on page 65 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	**DctAccessDone: DRAM controller access done**. 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.
30	**DctAccessWrite: DRAM controller read/write select**. Read-write. 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	**DctOffset: DRAM controller offset**. 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 section 2.8.8.4.8 [DRAM Address Timing and Output Driver Compensation Control] on page 75 for information on how to program this register.

Bits	Description
31:30	Reserved.
29:28	**ProcOdt: processor on-die termination**. Read-write. Cold reset: 00b. This field specifies the resistance of the on-die termination resistors. The definiton varies with the DDR type, F2x[1, 0]94[Ddr3Mode]. This field is valid only when F2x[1, 0]94[ProcOdtDis]=0. Bits DDR2 Definition — Bits DDR3 Definition 00b 300 ohms +/- 20% — 00b 240 ohms +/- 10% 01b 150 ohms +/- 20% — 01b 120 ohms +/- 10% 10b 75 ohms +/- 20% — 10b 60 ohms +/- 10% 11b Reserved — 11b Reserved
27:22	Reserved.

21:20	**DqsDrvStren: DQS drive strength**. Read-write. Cold reset: 11b. This field specifies the drive strength of the DQS pins. 00b 0.75x 10b 1.25x 01b 1.0x 11b 1.5x 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: • If all DIMMs of an unganged channel are populated with non-x4 devices, DataDrvStren is applied. • If the channels are ganged and populated with all non-x4 devices, DataDrvStren is applied. • If any DIMMs of an unganged channel are populated with x4 devices, DqsDrvStren is applied. • If the channels are ganged and populated with any x4 devices, DqsDrvStren is applied.
19:18	Reserved.
17:16	**DataDrvStren: data drive strength**. Read-write. Cold reset: 11b. This field specifies the drive strength of the DRAM data pins. 00b 0.75x. 10b 1.25x 01b 1.0x 11b 1.5x See the note in DqsDrvStren regarding how this field may be applied to DM signals as well.
15:14	Reserved.
13:12	**ClkDrvStren: MEMCLK drive strength**. Read-write. Cold reset: 11b. This field specifies the drive strength of the MEMCLK pins. 00b 1.0x. 10b 1.5x 01b 1.25x 11b 2.0x
11:10	Reserved.
9:8	**AddrCmdDrvStren: address/command drive strength**. Read-write. Cold reset: 11b. This field specifies the drive strength of the address, RAS, CAS, WE, bank and parity pins. 00b 1.0x. 10b 1.5x 01b 1.25x 11b 2.0x
7:6	Reserved.
5:4	**CsOdtDrvStren: CS/ODT drive strength**. Read-write. Cold reset: 11b. This field specifies the drive strength of the CS and ODT pins. 00b 1.0x. 10b 1.5x 01b 1.25x 11b 2.0x
3:2	Reserved.
1:0	**CkeDrvStren: CKE drive strength**. Read-write. Cold reset: 11b. This field specifies the drive strength of the CKE pins. 00b 1.0x. 10b 1.5x 01b 1.25x 11b 2.0x

\

F2x[1, 0]9C_x[302:301, 202:201, 102:101, 02:01] DRAM Write Data Timing [High:Low] Registers

See F2x[1, 0]98 for register access information.

These registers control the timing of write data with respect to MEMCLK and allow transmit DQS to be cen-

tered in the data eye. The delay starts 1 UI before the rising edge of MEMCLK corresponding to the CAS-write-latency. See section 2.8.8.8 [DRAM Training] on page 86 for information on how to use these registers.

Fine timing (WrDatFineDlyByte):
Delay = WrDatFineDlyByte * 1/64 of a MEMCLK, ranging from 0/64 to 31/64 MEMCLKs.

Gross timing (WrDatGrossDlyByte):
000b No delay
001b 0.5 MEMCLK delay
010b 1.0 MEMCLK delay
011b 1.5 MEMCLK delay
1xxb Reserved
Note: WrDatGrossDlyByte should be zero for DDR2 DIMMs. 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
31:29	**WrDatGrossDlyByte[7, 3]: write data gross delay byte[7, 3]**. Read-write. Reset: 0.
28:24	**WrDatFineDlyByte[7, 3]: write data fine delay byte[7, 3]**. Read-write. Cold reset: 0Fh.
23:21	**WrDatGrossDlyByte[6, 2]: write data gross delay byte[6, 2]**. Read-write. Reset: 0.
20:16	**WrDatFineDlyByte[6, 2]: write data fine delay byte[6, 2]**. Read-write. Cold reset: 0Fh.
15:13	**WrDatGrossDlyByte[5, 1]: write data gross delay byte[5, 1]**. Read-write. Reset: 0. Reset: 0.
12:8	**WrDatFineDlyByte[5, 1]: write data fine delay byte[5, 1]**. Read-write. Cold reset: 0Fh.
7:5	**WrDatGrossDlyByte[4, 0]: write data gross delay byte[4, 0]**. Read-write. Reset: 0. Reset: 0.
4:0	**WrDatFineDlyByte[4, 0]: write data fine delay byte[4, 0]**. Read-write. Cold reset: 0Fh.

F2x[1, 0]9C_x[303, 203, 103, 03] DRAM Write ECC Timing Register

See F2x[1, 0]98 for register access 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 section 2.8.8.8 [DRAM Training] on page 86 for information on how to use this register.

Bits	Description
31:8	Reserved.
7:5	**WrChkGrossDly: write data ECC gross delay**. Read-write. Reset: 0. This is encoded as follows: 000b No delay 001b 0.5 MEMCLK delay 010b 1.0 MEMCLK delay 011b 1.5 MEMCLK delay 1xxb Reserved
4:0	**WrChkFineDly: write data ECC fine delay**. 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

Cold reset: 0000 0000h. 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 section 2.8.8.4.8 [DRAM Address Timing and Output Driver Compensation Control] on page 75 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.

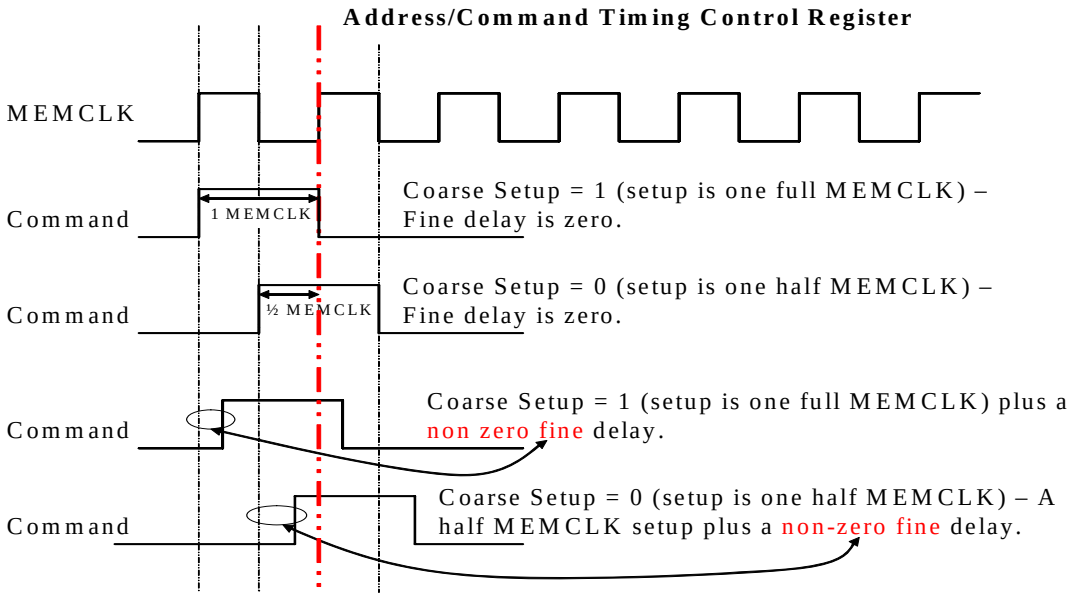

Figure 11: Address/Command Timing at the Processor Pins

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	**AddrCmdSetup: address/command setup time**. Read-write. 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	**AddrCmdFineDelay: address/command fine delay**. Read-write. 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	**CsOdtSetup: CS/ODT setup time**. Read-write. This bit selects the default setup time for the CS and ODT pins versus MEMCLK. 0b 1/2 MEMCLK 1b 1 MEMCLK
12:8	**CsOdtFineDelay: CS/ODT fine delay**. Read-write. 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	**CkeSetup: CKE setup time**. Read-write. This bit selects the default setup time for the CKE pins versus MEMCLK. 0b 1/2 MEMCLK 1b 1 MEMCLK
4:0	**CkeFineDelay: CKE fine delay**. Read-write. 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[306:305, 206:205, 106:105, 06:05] 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 timing of read (input) DQS signals with respect to data. See section 2.8.8.8 [DRAM Training] on page 86 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 with respect to data 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).

Bits	Description
31:30	Reserved.
29:24	**RdDqsTimeByte[7, 3]: read DQS byte [7, 3] timing control**. Read-write.
23:22	Reserved.
21:16	**RdDqsTimeByte[6, 2]: read DQS byte [6, 2] timing control**. Read-write.
15:14	Reserved.
13:8	**RdDqsTimeByte[5, 1]: read DQS byte [5, 1] timing control**. Read-write.
7:6	Reserved.
5:0	**RdDqsTimeByte[4, 0]: read DQS byte [4, 0] timing control**. Read-write.

F2x[1, 0]9C_x[307, 207, 107, 07] DRAM Read DQS ECC Timing Control Register

Cold reset: 0000 001Fh. See F2x[1, 0]98 for register access information. See section 2.8.8.8 [DRAM Training] on page 86 for information on how to use this register.

Bits	Description
31:6	Reserved.
5:0	**RdDqsTimeCheck: read DQS ECC byte timing control**. 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, F2x[1, 0]94[MemClkFreq], as follows: • For memory clock frequencies less than 400 MHz, delay = RdDqsTimeCheck * 1/128 MEMCLKs, ranging from 0 to 63/128 MEMCLKs. • For memory clock frequencies of 400 MHz or greater, delay = (RdDqsTimeCheck & 03Eh) * 1/128 MEMCLKs, ranging from 0 to 62/128 MEMCLKs (writes to the LSB of RdDqsTimeByte are ignored and reads return zero).

F2x[1, 0]9C_x08 DRAM Phy Control Register

Cold reset: 0208 0000h. See F2x[1, 0]98 for register access information. See section 2.8.8.8 [DRAM Training] on page 86 for information on how to use this register.

Bits	Description
31	Reserved.
30	**DisAutoComp: disable automatic compensation**. Read-write. 1=Disable the compensation control state machine. 0=The phy automatic compensation engine is enabled.
29:14	Reserved.
13	**DqsRcvTrEn: DQS receiver training enable**. 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	**WrLvOdtEn: write levelization ODT enabled**. Read-write. 1=ODT enabled during write levelization training. 0=No ODT is used for write levelization training.

11:8	**WrLvOdt[3:0]: write levelization ODT**. 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. Note: tri-state enable for ODT is turned off by the phy while WrLvOdtEn is set. Bit / Fr2(1207) pin name / AM2r2 pin name / AM3 pin name 0 / M[B, A]0_ODT[0] / M[B, A]0_ODT[0] / M[B, A]0_ODT[0] 1 / M[B, A]1_ODT[0] / M[B, A]1_ODT[0] / M[B, A]1_ODT[0] 2 / M[B, A]2_ODT[0] / N/A / M[B, A]0_ODT[1] 3 / M[B, A]3_ODT[0] / N/A / M[B, A]1_ODT[1]
7:6	Reserved.
5:4	**TrDimmSel: training DIMM select**. 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 [The DRAM CS Base Address Registers] F2x[1, 0][5C:40]. For DDR3, bit[5] is reserved.
3	**PhyFenceTrEn: phy fence training enable**. Write-only. 1=Initiate phy based fence training. 0=Stop the phy based fence training engine.
2	**TrNibbleSel: training nibble select**. Read-write. This specifies nibbles of each DIMM data and ECC byte trained during write levelization training. 0=Lower nibbles. 1=Upper nibbles.
1	**WrtLvTrMode: write levelization training mode**. Read-write. 1=Write levelization training is done by the BIOS. 0=Write training is done by the hardware.
0	**WrtLvTrEn: write levelization training enable**. 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 xxxxh. 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 section 2.8.8.2 [Phy compensation initialization] on page 70 for more information on how to program these registers.

Note: BIOS must not write to this register.

Bits	Description
31:30	Reserved.
29:25	**D3Cmp2DrvPCal: D3CMP 2 driver PMOS calibration code**. Read-write.
24:20	**D3Cmp2DrvNCal: D3CMP 2 driver NMOS calibration code**. Read-write.
19:15	**D3Cmp1DrvPCal: D3CMP 1 driver PMOS calibration code**. Read-write.
14:10	**D3Cmp1DrvNCal: D3CMP 1 driver NMOS calibration code**. Read-write.
9:5	**D3Cmp0DrvPCal: D3CMP 0 driver PMOS calibration code**. Read-write.
4:0	**D3Cmp0DrvNCal: D3CMP 0 driver NMOS calibration code**. Read-write.

F2x9C_x0A DRAM Phy Predriver Calibration Register

Cold reset: 0631 8C63h. See F2x[1, 0]98 for register access information. See section 2.8.8.2 [Phy compensation initialization] on page 70 for information on how to program this register.

Bits	Description

<table>
<tr><td>31:28</td><td>Reserved.</td></tr>
<tr><td>27:25</td><td>D3Cmp2PCal: D3CMP 2 PMOS predriver calibration code. Read-write. The 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.

DataDrvStren Calibration Code
00b 7h
01b 7h
10b 5h
11b 3h</td></tr>
<tr><td>24:23</td><td>Reserved.</td></tr>
<tr><td>22:20</td><td>D3Cmp2NCal: D3CMP 2 NMOS predriver calibration code. Read-write. The 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.

DataDrvStren Calibration Code
00b 7h
01b 7h
10b 3h
11b 2h</td></tr>
<tr><td>19:18</td><td>Reserved.</td></tr>
<tr><td>17:15</td><td>D3Cmp1PCal: D3CMP 1 PMOS predriver calibration code. Read-write. The 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.

AddrCmdDrvStren Calibration Code
00b 7h
01b 5h
10b 3h
11b 2h</td></tr>
<tr><td>14:13</td><td>Reserved.</td></tr>
<tr><td>12:10</td><td>D3Cmp1NCal: D3CMP 1 NMOS predriver calibration code. Read-write. The 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.

AddrCmdDrvStren Calibration Code
00b 7h
01b 3h
10b 2h
11b 2h</td></tr>
<tr><td>9:8</td><td>Reserved.</td></tr>
</table>

7:5	**D3Cmp0PCal: D3CMP 0 PMOS predriver calibration code**. Read-write. The 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. DataDrvStren — Calibration Code 00b — 7h 01b — 7h 10b — 5h 11b — 3h
4:3	Reserved.
2:0	**D3Cmp0NCal: D3CMP 0 NMOS predriver calibration code**. Read-write. The 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. DataDrvStren — Calibration Code 00b — 7h 01b — 7h 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. Based on the system configuration, BIOS may tri-state signals with associated chip selects that are unpopulated in an effort to conserve power. 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 section 2.8.8.6 [Phy Fence programming] on page 85 for information on how to program this register.

Bits	Description
31:21	Reserved.
20:16	**PhyFence: phy fence**.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.
15:14	Reserved.
13:12	**CKETri[1:0]: CKE tri-state**. 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 CKETri[1:0] are mapped to packages as follows: Bit — Package pin name 0 — M[B, A]_CKE[0] 1 — M[B, A]_CKE[1] Note: F2x9C_x0C controls the channel A memory CKE pins and F2x19C_x0C controls the channel B memory CKE pins.

<table>
<tr><td>11:8</td><td>ODTTri[3:0]: ODT tri-state. Read-write. 0=The ODT signals are not tri-stated unless directed to by the DCT. 1=Tri-state unconnected ODT signals from the processor. The bits ODTTri[3:0] are mapped to packages as follows:
<table>
<tr><th>Bit</th><th>Fr2(1207) pin name</th><th>AM2r2 pin name</th><th>AM3 pin name</th></tr>
<tr><td>0</td><td>M[B, A]0_ODT[0]</td><td>M[B, A]0_ODT[0]</td><td>M[B, A]0_ODT[0]</td></tr>
<tr><td>1</td><td>M[B, A]1_ODT[0]</td><td>M[B, A]1_ODT[0]</td><td>M[B, A]1_ODT[0]</td></tr>
<tr><td>2</td><td>M[B, A]2_ODT[0]</td><td>N/A</td><td>M[B, A]0_ODT[1]</td></tr>
<tr><td>3</td><td>M[B, A]3_ODT[0]</td><td>N/A</td><td>M[B, A]1_ODT[1]</td></tr>
</table>
Note: F2x9C_x0C controls the channel A memory ODT pins and F2x19C_x0C controls the channel B memory ODT pins.</td></tr>
<tr><td>7:0</td><td>ChipSelTri[7:0]: chip select tri-state. Read-write. 0=The chip select signals are not tri-stated unless directed to by the DCT. 1=Tri-state unpopulated chip selects when motherboard termination is available. 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 bits ChipSelTri[7:0] are mapped to packages as follows:
<table>
<tr><th>Bit</th><th>Fr2(1207) pin name</th><th>AM2r2 pin name</th><th>AM3 pin name</th></tr>
<tr><td>0</td><td>M[B, A]0_CS_H/L[0]</td><td>M[B, A]0_CS_H/L[0]</td><td>M[B, A]0_CS_H/L[0]</td></tr>
<tr><td>1</td><td>M[B, A]0_CS_H/L[1]</td><td>M[B, A]0_CS_H/L[1]</td><td>M[B, A]0_CS_H/L[1]</td></tr>
<tr><td>2</td><td>M[B, A]1_CS_H/L[0]</td><td>M[B, A]1_CS_H/L[0]</td><td>M[B, A]0_CS_H/L[2]</td></tr>
<tr><td>3</td><td>M[B, A]1_CS_H/L[1]</td><td>M[B, A]1_CS_H/L[1]</td><td>M[B, A]0_CS_H/L[3]</td></tr>
<tr><td>4</td><td>M[B, A]2_CS_H/L[0]</td><td>N/A</td><td>N/A</td></tr>
<tr><td>5</td><td>M[B, A]2_CS_H/L[1]</td><td>N/A</td><td>N/A</td></tr>
<tr><td>6</td><td>M[B, A]3_CS_H/L[0]</td><td>N/A</td><td>N/A</td></tr>
<tr><td>7</td><td>M[B, A]3_CS_H/L[1]</td><td>N/A</td><td>N/A</td></tr>
</table>
Note: F2x9C_x0C controls the channel A memory chip select pins and F2x19C_x0C controls the channel B memory chip select pins.</td></tr>
</table>

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.

DctOffset	Register
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 from the start of the read preamble with respect to MEMCLK. See section 2.8.8.8 [DRAM Training] on page 86 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
0001b	0.5 MEMCLK delay
0010b	1.0 MEMCLK delay
0011b	1.5 MEMCLK delay
0100b	2.0 MEMCLK delay
0101b	2.5 MEMCLK delay
0110b	3.0 MEMCLK delay
0111b	3.5 MEMCLK delay
1xxxb	Reserved.

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 Dct-Offset 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	**DqsRcvEnGrossDelay[7, 5],[3, 1]: DQS receiver enable gross delay[7, 5],[3, 1]**. Read-write. Reset: 0.
20:16	**DqsRcvEnFineDelay[7, 5],[3, 1]: DQS receiver enable fine delay[7, 5],[3, 1]**. Read-write. Cold reset: 0.
15:13	Reserved.
12:9	Reserved.
8:5	**DqsRcvEnGrossDelay[6, 4],[2, 0]: DQS receiver enable gross delay[6, 4],[2, 0]**. Read-write. Reset: 0.
4:0	**DqsRcvEnFineDelay[6, 4],[2, 0]: DQS receiver enable fine delay[6, 4],[2, 0]**. 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	**DqsRcvEnGrossDelayCheck: DQS receiver enable gross delay ECC**. Read-write. Reset: 0.
4:0	**DqsRcvEnFineDelayCheck: DQS receiver enable fine delay ECC**.Read-write. Cold reset: 0.

F2x[1, 0]9C_x[45:30] 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 two groups of registers for each DDR3 DIMM. DIMM numbers are specified by [The DRAM CS Base Address Registers] F2x[1, 0][5C:40].

DctOffset	Register
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 - 3Fh	Reserved
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

Each of these registers control the DQS timing delay for write commands relative to MEMCLK. See section 2.8.8.8 [DRAM Training] on page 86 for information on how to use this register. 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_0045h , 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	**WrDqsGrossDly[7, 5],[3, 1]: DQS write gross delay[7, 5],[3, 1]**. Read-write. Reset: 0.
20:16	**WrDqsFineDly[7, 5],[3, 1]: DQS write fine delay[7, 5],[3, 1]**. Read-write. Cold reset: 0.
15	Reserved.
14:13	Reserved.
12:8	Reserved.
7:5	**WrDqsGrossDly[6, 4],[2, 0]: DQS write gross delay[6, 4],[2, 0]**. Read-write. Reset: 0.
4:0	**WrDqsFineDly[6, 4],[2, 0]: DQS write fine delay[6, 4],[2, 0]**. 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	**WrDqsChkGrossDly: DQS write gross delay ECC**. Read-write. Reset: 0.
4:0	**WrDqsChkFineDly: DQS write fine delay ECC**. 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 section 2.8.8.8 [DRAM Training] on page 86 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	**PhRecGrossDlyByte[7, 3]: phase recovery gross delay byte [7, 3]**. Read-write. Reset: X.
28:24	**PhRecFineDlyByte[7, 3]: phase recovery fine delay byte [7, 3]**. Read-write. Reset: X.
23	Reserved.
22:21	**PhRecGrossDlyByte[6, 2]: phase recovery gross delay byte [6, 2]**. Read-write. Reset: X.
20:16	**PhRecFineDlyByte[6, 2]: phase recovery fine delay byte [6, 2]**. Read-write. Reset: X.
15	Reserved.
14:13	**PhRecGrossDlyByte[5, 1]: phase recovery gross delay byte [5, 1]**. Read-write. Reset: X.
12:8	**PhRecFineDlyByte[5, 1]: phase recovery fine delay byte [5, 1]**. Read-write. Reset: X.
7	Reserved.
6:5	**PhRecGrossDlyByte[4, 0]: phase recovery gross delay byte [4, 0]**. Read-write. Reset: X.
4:0	**PhRecFineDlyByte[4, 0]: phase recovery fine delay byte [4, 0]**. 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 is 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	**PhRecEccGrossDlyByte: phase recovery ECC gross delay byte**. Read-write. Reset: X.
4:0	**PhRecEccFineDlyByte: phase recovery ECC fine delay byte**. 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 section 2.8.8.8 [DRAM Training] on page 86 for information on how to use this register.

Bits	Description
31:9	Reserved.
8:0	**WrLvErr: write levelization error**. 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]A0 DRAM Controller Miscellaneous Register

Reset: 0000 0000h. See section 2.8.1 [DCT Configuration Registers] on page 65 for general programming information about DCT configuration registers.

Bits	Description
31:10	Reserved.
9	**DramEnabled: DRAM enabled**. Read-only. This bit is identical to F2x110[DramEnabled].
8:1	Reserved.
0	**MemCleared: memory cleared**. Read-only. This bit is identical to F2x110[MemCleared].

F2x[1, 0]A8 DRAM Controller Miscellaneous Register 2

Reset: 0000 0000h. See section 2.8.1 [DCT Configuration Registers] on page 65 for general programming information about DCT configuration registers.

Bits	Description
31:16	Reserved.
15:8	**CtrlWordCS[7:0]: control word chip select**. 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]. 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.
7:0	Reserved.

F2x110 DRAM Controller Select Low Register

Reset: 0000 0000h.

Bits	Description
31:11	**DctSelBaseAddr[47:27]: DRAM controller select base address bits[47:27]**. 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	**MemCleared: memory cleared**. Read-only. 1=Memory has been cleared since the last warm reset. This bit is set by MemClrInit. See MemClrInit below.
9	**MemClrBusy: memory clear busy**. Read-only. 1=The memory clear operation in either of the DCTs is in progress. Reads or writes to DRAM while the memory clear operation is in progress result in undefined behavior.
8	**DramEnable: DRAM enabled**. Read-only. 1=All of the used DCTs are initialized (see section 2.8.8.5 [DRAM Device Initialization] on page 79) or have exited from self refresh (F2x[1, 0]90[ExitSelfRef] transitions from 1 to 0).

7:6	**DctSelIntLvAddr: DRAM controller select channel interleave address bit**. 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 functions are: 00b Address bit 6. 10b Hash: exclusive OR of address bits[20:16, 6]. 01b Address bit 12. 11b Reserved
5	**DctDatIntLv: DRAM controller data interleave enable**. 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 is clear), DctDatIntLv should be enabled. See section 2.13.2 [DRAM Considerations for ECC] on page 120 for more information.
4	**DctGangEn: DRAM controller ganging enable**. Read-write. 1=Both DCTs are ganged to form a single double-width DDR interface. 0=The DCTs operate independently. This also affects how DCT configuration registers; see section 2.8.1 [DCT Configuration Registers] on page 65. Note, if ganging is to be enabled, this bit must be set prior to programming any DCT registers.
3	**MemClrInit: memory clear initialization**. 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 status of the memory clear operation can be determined by reading the MemClrBusy and MemCleared bits. This command is ignored if MemClrBusy=1 when the command is received. 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.
2	**DctSelIntLvEn: DRAM controller interleave enable**. 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.
1	**DctSelHi: DRAM controller high select**. 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.
0	**DctSelHiRngEn: DRAM controller select high range enable**. 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.

F2x114 DRAM Controller Select High Register

Reset: 0000 0000h.

Bits	Description
31:10	**DctSelBaseOffset[47:26]: DRAM controller select base offset address bits[47:26]**. Read-write. When F2x110[DctSelHiRngEn]=1, this value is subtracted from the physical address of certain transactions before being passed to the DCT. See section 2.8.10.2 [DctSelBaseOffset Programming] on page 106 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	**MctVarPriCntLmt: variable priority time limit**. Read-write. Reset: 0000b. 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	**McqHiPriByPassMax: memory controller high priority bypass max**. 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	**McqMedPriByPassMax: memory controller medium bypass low priority max**. 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	**MctPriScrub: scrubber priority**. Read-write. Reset: medium (00b).
15:14	**MctPriTrace: trace-mode request priority**. Read-write. Reset: high (10b). This must be set to high.
13:12	**MctPriIsoc: display refresh read priority**. Read-write. Reset: high (10b). See 2.6.4.2.4 [F0x[5C:40]Display Refresh And IFCM] on page 57.
11:10	**MctPriWr: default write priority**. Read-write. Reset: low (01b).
9:8	**MctPriDefault: default non-write priority**. Read-write. Reset: medium (00b).
7:6	**MctPriIsocWr: IO write with the isoch bit set priority**. Read-write. Reset: medium (00b). This does not apply to isochronous traffic that is classified as display refresh.
5:4	**MctPriIsocRd: IO read with the isoch bit set priority**. Read-write. Reset: high (10b). This does not apply to isochronous traffic that is classified as display refresh.
3:2	**MctPriCpuWr: CPU write priority**. Read-write. Reset: low (01b).
1:0	**MctPriCpuRd: CPU read priority**. 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 depending on the confidence. In addition, a fixed stride mode (non-alternating) may be used for IO requests which often have fixed stride patterns. This mode bypasses the stride predictor such that CPU-access stride predictions are not adversely affected by IO streams.

The MCT tracks several stride patterns simultaneously. Each of these has a confidence level associated with it that varies as follows:

- Each time a request is received that matches the stride pattern, the confidence level increases by one.
- Each time a request is received within +/- 4 cachelines of the last requested cacheline in the pattern that does not match the pattern, then the confidence level decreases by one.
- When the confidence level reaches the saturation point specified by PrefConfSat, then it no-longer increments.

Each request that is not within +/- 4 cachelines of the last requested cacheline line of all the stride patterns tracked initiates a new stride pattern by displacing one of the existing least-recently-used stride patterns.

- The following settings should be used:

PrefThreeConf = 7	PrefTwoConf = 7	PrefOneConf = 2
PrefConfSat = 1	MctWrLimit = 16	DctWrLimit = 0

Note: BIOS should enable prefetching by clearing F2x11C[PrefIoDis] and F2x11C[PrefCpuDis].

Bits	Description
31	**MctScrubEn: MCT scrub enable**. 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	**FlushWr: flush writes command**. 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	**FlushWrOnStpGnt: flush writes on stop-grant**. 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	**PrefDramTrainMode: prefetch DRAM training mode**. 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 section 2.8.8.8.5 [Continuous Pattern Generation] on page 100.
27:25	**PrefThreeConf: prefetch three-ahead confidence**. Read-write. Reset: 110b. Confidence level required in order to prefetch three cachelines ahead (same encoding as PrefTwoConf below).
24:22	**PrefTwoConf: prefetch two-ahead confidence**. Read-write. Reset: 011b. Confidence level required in order to prefetch two cachelines ahead. 000b = 0 001b = 2 ... 111b = 14
21:20	**PrefOneConf: prefetch one-ahead confidence**. Read-write. Reset: 10b. Confidence level required in order to prefetch one ahead (0 through 3).
19:18	**PrefConfSat: prefetch confidence saturation**. 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	**PrefFixDist: prefetch fixed stride distance**. 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	**PrefFixStrideEn: prefetch fixed stride enable**. Read-write. Reset: 0. 1=The prefetch stride for all requests (CPU and IO) is fixed (non-alternating).
14	**PrefIoFixStrideEn: Prefetch IO fixed stride enable**. Read-write. Reset: 0. 1=The prefetch stride for IO requests is fixed (non-alternating).
13	**PrefIoDis: prefetch IO-access disable**. Read-write. Reset: 1. 1=Disables IO requests from triggering prefetch requests.
12	**PrefCpuDis: prefetch CPU-access disable**. Read-write. Reset: 1. 1=Disables CPU requests from triggering prefetch requests.
11:7	**MctPrefReqLimit: memory controller prefetch request limit**. Read-write. Reset: 1Fh (31). Specifies the maximum number of outstanding prefetch requests allowed. See F3x78 for restrictions on this field.

6:2	**MctWrLimit: memory controller write-burst limit**. 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	**DctWrLimit: DRAM controller write limit**. 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. 00b = 0 01b = 1 10b = 2 11b = no limit.

3.6 Function 3 Miscellaneous Control Registers

See section 3.1 [Register Descriptions and Mnemonics] on page 137 for a description of the register naming convention. See section 2.11 [Configuration Space] on page 113 for details about how to access this space.

F3x00 Device/Vendor ID Register

Reset: 1203 1022h.

Bits	Description
31:16	**DeviceID: device ID**. Read-only.
15:0	**VendorID: vendor ID**. Read-only.

F3x04 Status/Command Register

Reset: 0000 0000h, except bit[20]; see below.

Bits	Description
31:16	**Status**. Read-only. Bit[20] is set to indicate the existence of a PCI-defined capability block, if one exists.
15:0	**Command**. Read-only.

F3x08 Class Code/Revision ID Register

Reset: 0600 0000h.

Bits	Description
31:8	**ClassCode**. Read-only. Provides the host bridge class code as defined in the PCI specification.
7:0	**RevID: revision ID**. Read-only.

F3x0C Header Type Register

Reset: 0080 0000h.

Bits	Description
31:0	**HeaderTypeReg**. Read-only. These bits are fixed at their default values. The header type field indicates that there are multiple functions present in this device.

F3x34 Capability Pointer Register

Reset: 0000 00??h.

Bits	Description
31:8	Reserved.
7:0	**CapPtr**. 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.

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	**TblWlkDatErrEn: table walk data error enable**. Read-write. 1=Enables MCA reporting of uncorrectable errors in returned data from a DEV or GART table walk.
26	**NbArrayParEn: Northbridge array parity error reporting enable**. Read-write. 1=Enables MCA reporting of parity errors in the NB arrays.
25	**McaUsPwDatErrEn: MCA upstream data error enable**. Read-write. 1=Enables MCA reporting of upstream posted writes in which the link error bits indicate a data error.
24	**SyncPkt3En: link 3 sync packet error reporting enable**. 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	**CrcErr3En: link 3 CRC error reporting enable**. Read-write. 1=Enables MCA reporting of CRC errors detected on link 3 (see the description of CRC Error in Table 46). 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	**RtryHt3En: link 3 retry reporting enable**. Read-write. 1=Enables MCA reporting of retries on link 3.
21	**RtryHt2En: link 2 retry reporting enable**. Read-write. 1=Enables MCA reporting of retries on link 2.

20	**RtryHt1En: link 1 retry reporting enable**. Read-write. 1=Enables MCA reporting of retries on link 1.
19	**RtryHt0En: link 0 retry reporting enable**. Read-write. 1=Enables MCA reporting of retries on link 0.
18	**DramParEn: DRAM parity error reporting enable**. Read-write. 1=Enables MCA reporting of parity errors on the DRAM address or control signals.
17	**HtDataEn: link data error reporting enable**. Read-write. 1=Enables MCA reporting of packets with data errors detected on links.
16	**ProtEn: protocol error reporting enable**. 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	**L3ArrayUCEn: L3 cache array uncorrectable error reporting enable**. Read-write. 1=Enables MCA reporting of uncorrectable errors in the L3 cache arrays.
14	**L3ArrayCorEn: L3 cache array correctable error reporting enable**. Read-write. 1=Enables MCA reporting of correctable errors in the L3 cache arrays.
13	**DevErrEn: DEV error reporting enable**. Read-write. 1=Enables MCA reporting of SVM DEV errors.
12	**WDTRptEn: watchdog timer error reporting enable**. Read-write. 1=Enables MCA reporting of watchdog timer errors. The watchdog timer checks for NB system accesses for which a response is expected but no response is received. See [The MCA NB Configuration Register] F3x44 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	**AtomicRMWEn: atomic read-modify-write error reporting enable**. 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	**GartTblWkEn: GART table walk error reporting enable**. Read-write. 1=Enables MCA reporting of GART cache table walks which encounter a GART PTE entry which is invalid.
9	**TgtAbortEn: target abort error reporting enable**. Read-write. 1=Enables MCA reporting of target aborts to a link. The NB returns an error response back to the requestor with any associated data all 1s independent of the state of this bit.
8	**MstrAbortEn: master abort error reporting enable**. Read-write. 1=Enables MCA reporting of master aborts to a link. The NB returns an error response back to the requestor with any associated data all 1s independent of the state of this bit.
7	**SyncPkt2En: link 2 sync packet error reporting enable**. 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	**SyncPkt1En: link 1 sync packet error reporting enable**. 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	**SyncPkt0En: link 0 sync packet error reporting enable**. 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	**CrcErr2En: link 2 CRC error reporting enable**. Read-write. 1=Enables MCA reporting of CRC errors detected on link 2 (see the description of CRC Error in Table 46). 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	**CrcErr1En: link 1 CRC error reporting enable**. Read-write. 1=Enables MCA reporting of CRC errors detected on link 1 (see the description of CRC Error in Table 46). 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	**CrcErr0En: link 0 CRC error reporting enable**. Read-write. 1=Enables MCA reporting of CRC errors detected on link 0 (see the description of CRC Error in Table 46). 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	**UECCEn: uncorrectable ECC error reporting enable**. 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	**CECCEn: correctable ECC error reporting enable**. 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 also [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	**NbMcaLogEn: Northbridge MCA log enable**. Read-write. 1=Enables logging (but not reporting) of NB MCA errors even if MCA is not globally enabled.
30	**SyncOnDramAdrParErrEn: sync flood on DRAM address parity error enable**. Read-write. 1=Enables sync flood on detection of a DRAM address parity error.
29	**DisMstAbtCpuErrRsp: master abort CPU error response disable**. Read-write. 1=Disables master abort reporting through the CPU MCA error-reporting banks.
28	**DisTgtAbtCpuErrRsp: target abort CPU error response disable**. Read-write. 1=Disables target abort reporting through the CPU MCA error-reporting banks.

27	**NbMcaToMstCpuEn: machine check errors to master CPU only**. 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 (MSR0000_0410, MSR0000_0411, MSR0000_0412, MSR0000_0413, MSRC000_04[0A:08], MSRC001_0048) 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 section 3.1 [Register Descriptions and Mnemonics] on page 137 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: • When the CPU which originated the request is known, it is stored in F3x4C[ErrCPU], regardless of the setting of NbMcaToMstCpuEn. See Table 48 for errors where ErrCPU is known. • If IO originated the request, then the error is reported to core 0, regardless of the setting of NbMcaToMstCpuEn. • BIOS should set this bit to 1 in all processors if the machine check handler must execute on core 0.
26	**CorrMcaExcEn: correctable error MCA exception enable**. Read-write. 1=Correctable errors that are enabled for checking and logging cause a machine check exception (reporting) in addition to being logged.
25	**DisPciCfgCpuErrRsp: PCI configuration CPU error response disable**. Read-write. 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 also F3x180[DisPciCfgCpuMstAbtRsp], which applies only to master aborts.
24	**IoRdDatErrEn: IO read data error log enable**. Read-write. 1=Enables logging and reporting of read data errors (link defined master aborts, target aborts, and data error) for data destined for IO devices. 0=Read data errors for transactions from IO devices are not logged by MCA, although error responses may still be generated to the requesting IO device.
23	**ChipKillEccEn: chipkill ECC mode enabled**. Read-only. 1=Chipkill ECC mode is enabled; ECC checking is based on ganged 128/16-bit data/ECC and can be used for chipkill. 0=Chipkill ECC mode is not enabled; ECC checking is based on two interleaved, unganged 64/8-bit data/ECC lines and cannot be used for chipkill. Chipkill functionality is possible only when chipkill ECC mode is enabled (as indicated in this field) and the physical configuration is appropriate; see section 2.13.2 [DRAM Considerations for ECC] on page 120 for more details.
22	**DramEccEn: DRAM ECC enable**. Read-write. 1=Enables ECC check/correct mode. This bit must be set in order for ECC checking/correcting by the NB to be enabled. If set, ECC is checked and correctable errors are corrected irrespective of whether machine check ECC reporting is enabled. The hardware only allows values to be programmed into this field which are consistent with the ECC capabilities of the device as specified in [The Northbridge Capabilities Register] F3xE8. Attempts to write values inconsistent with the capabilities result in this field not being updated. This bit does not affect ECC checking in the Northbridge arrays.
21	**SyncOnAnyErrEn: sync flood on any error enable**. Read-write. 1=Enables flooding of all links with sync packets on detection of any MCA error that is uncorrectable, including Northbridge array errors and link protocol errors.
20	**SyncOnWDTEn: sync flood on watchdog timer error enable**. Read-write. 1=Enables flooding of all links with sync packets on detection of a watchdog timer error. BIOS should set this bit to 1b.

19:18	**SubLinkSel: sublink select for CRC error generation**. Read-write. Selects the sublink of a link selected by LdtLinkSel to be used for CRC error injection through GenCrcErrByte0 and GenCrcErrByte1. When the link is ganged, SubLinkSel must be 00b. When the link is unganged, the following values indicate which sublink is selected: 00b = Sublink 0. 10b = Reserved. 01b = Sublink 1. 11b = Reserved.
17	**GenCrcErrByte1: generate CRC error on byte lane 1**. Read-Write. 1=For ganged links, causes a CRC error to be injected on byte lane 1 of the link specified by LdtLinkSel. For unganged links, this field is reserved, and GenCrcErrByte0 must be used. The data carried by the link is unaffected. This bit is cleared after the error has been generated. See also F0x150[ForceErrType].
16	**GenCrcErrByte0: generate CRC error on byte lane 0**. Read-Write. 1=Causes a CRC error to be injected on byte lane 0 of the link specified by LdtLinkSel and the sublink specified by SubLinkSel. The data carried by the link is unaffected. This bit is cleared after the error has been generated. See also F0x150[ForceErrType].
15:14	**LdtLinkSel: link select for CRC error generation**. Read-Write. Selects the link to be used for CRC error injection through GenCrcErrByte1/GenCrcErrByte0. 00b = link 0 01b = link 1 10b = link 2 11b = link 3
13:12	**WDTBaseSel: watchdog timer time base select**. Read-write. Selects the time base used by the watchdog timer. The counter selected by WDTCntSel determines the maximum count value in the time base selected by WDTBaseSel. 00b = 1.31 ms 10b = 5 ns 01b = 1.28 us 11b = reserved
11:9	**WDTCntSel[2:0]: watchdog timer count select bits[2:0]**. 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: 0000b = 4095 0100b = 255 1000b = 8191 0001b = 2047 0101b = 127 1001b = 16383 0010b = 1023 0110b = 63 1010b - 1111b = reserved 0011b = 511 0111b = 31 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.
8	**WDTDis: watchdog timer disable**. 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.
7	**IoErrDis: IO error response disable**. 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.
6	**CpuErrDis: CPU error response disable**. Read-write. 1=Disables generation of a read data error response to the core on detection of a target or master abort error condition.

5	**IoMstAbortDis: IO master abort error response disable**. 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.
4	**SyncPktPropDis: sync packet propagation disable**. 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.
3	**SyncPktGenDis: sync packet generation disable**. Read-write. 1=Disables flooding of all outgoing links with sync packets when a CRC error is detected on an incoming link. By default, sync packet generation for CRC errors is controlled through [The Link Control Registers] F0x[E4, C4, A4, 84].
2	**SyncOnUcEccEn: sync flood on uncorrectable ECC error enable**. Read-write. 1=Enables flooding of all links with sync packets on detection of an uncorrectable ECC error.
1	**CpuRdDatErrEn: CPU read data error log enable**. Read-write. 1=Enables logging and reporting of read data errors (master aborts and target aborts) for data destined for the CPU on this node. This bit should be clear if read data error logging is enabled for the remaining error reporting blocks in the CPU. Logging the same error in more than one block may cause a single error event to be treated as a multiple error event and cause the CPU to enter shutdown.
0	Reserved.

F3x48 MCA NB Status Low Register

Cold reset: xxxx xxxxh.
Software is normally only allowed to write 0's to this register to clear the fields so subsequent errors may be logged. See also 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	**Syndrome[15:8]: syndrome bits 15:8 for ECC**. Read-write. Logs the upper eight syndrome bits when an ECC error is detected.
23:21	Reserved.
20:16	**ErrorCodeExt: extended error code**. Read-write. Logs the extended error code when an error is detected. See Table 47 for encoding.
15:0	**ErrorCode: error code**. Read-write. Logs an error code when an error is detected. See Table 47 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

Error Code	Error Code Type	Description
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 41: Error codes: transaction type

TT	Transaction Type
00	Instruction
01	Data
10	Generic
11	Reserved

Table 42: Error codes: cache level

LL	Cache Level
00	Reserved
01	Level 1 (L1)
10	Level 2 (L2)
11	Generic (LG; includes L3 cache)

Table 43: 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 44: 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 45: Error codes: memory or IO

II	Memory or IO
00	Memory Access (MEM)
01	Reserved
10	IO Access (IO)
11	Generic (GEN)

Table 46: 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

Table 46: NB error descriptions

Error Type	Description	Control Bits (F3x40)
ECC Error	DRAM ECC error detected in the NB.	CECCEn, UECCEn
DEV Error	SVM DEV error detected.	DevErrEn
Link Data Error	Data error detected on link.	HtDataEn, McaUsPwDatErrEn
Protocol Error	Protocol error detected by linkor L3. These errors are distinguished from each other by the value in MSR0000_0412[ErrAddr]. See Table 50. For protocol errors, the system cannot continue operation. 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
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 Table 48 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 47: 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[1]	LG
Tgt Abort		0_0100	BUS	SRC/OBS	0	RD/WR	MEM/IO[1]	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	LG
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/-[2]	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
L3 Cache Data Error	L3 Cache	1_1100	MEM	-	-	RD/Evict /Snoop /GEN	GEN	LG
L3 CacheTag Error		1_1101	MEM	-	-	RD/Evict /Snoop /GEN	GEN	LG
L3 Cache LRU Error		1_1110	MEM	-	-	RD/Evict /Snoop /GEN	GEN	LG

1. Indicates the type of link attached to the reporting NB, not the instruction type. MEM indicates coherent link, IO indicates IO link.
2. Error thresholding group is Link if link protocol error, none if L3 protocol error.

Table 48: NB error signatures, part 2

Error Type	F3x4C settings									
	29 UC	**26 AddrV**	**25 PCC**	**Syndrome Valid**	**14 CECC**	**13 UECC**	**12 Reserved**	**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[1]	1	-	0	0		0	-	-
ECC Error	If multi-symbol[8]	1	If multi-symbol[8] and CPU source	15:0	If not multi-symbol[8]	If multi-symbol[8]		1/0	-	-
DEV Error	1	1	0	-	0	0		0	Y	-
Link Data Error	1	1	0	-	0	0		0	Y	-
Protocol Error	1	1/0[2]	1	-	0	0		0	Y[3]	-
NB Array Error	1	1[4]	1	-	0	0		0	-	-
DRAM Parity Error	1	0	1	-	0	0		0	-	-
Link Retry[9]	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 53, “MCA NB Address Low Register encoding for Watchdog Timer Errors,” on page 223
2. See Table 50, “MCA NB Address Low Register encoding Protocol Errors,” on page 221
3. Link identified only if link protocol error. See entry in Table 46 for details.
4. See Table 51, “MCA NB Address Low Register encoding for NB Array Errors,” on page 222
5. See Table 52, “MCA NB Address Register encoding for L3 Array Errors,” on page 223
6. This field contains the L3 way in error. F3x4C[McaStatSubCache] contains the subcache number.
7. Depends on Memory Transaction Type (Table 43); valid if non-zero.
8. x4 Chipkill ECC is always enabled in ganged mode. See F3x44[ChipKillEccEn] for information on symbol size.
9. Retries initiated by either side of the link are logged.
10.

Table 48: NB error signatures, part 2

Error Type	F3x4C settings									
	29 UC	**26 AddrV**	**25 PCC**	**Syndrome Valid**	**14 CECC**	**13 UECC**	**12 Reserved**	**8 Scrub**	**7:4 LDT Link**	**3:0 Err CPU**
DEV Table Walk Data Error	1	1	0	-	0	0		0	-	-
L3 Cache Data Error	If UECC	1[5]	If UC	15:0	If single-bit	If multi-bit		1/0	Y[6]	Y[7]
L3 Cache Tag Error										
L3 Cache LRU Error	0	1[5]	0	-	0	0		0	-	-

1. See Table 53, “MCA NB Address Low Register encoding for Watchdog Timer Errors,” on page 223
2. See Table 50, “MCA NB Address Low Register encoding Protocol Errors,” on page 221
3. Link identified only if link protocol error. See entry in Table 46 for details.
4. See Table 51, “MCA NB Address Low Register encoding for NB Array Errors,” on page 222
5. See Table 52, “MCA NB Address Register encoding for L3 Array Errors,” on page 223
6. This field contains the L3 way in error. F3x4C[McaStatSubCache] contains the subcache number.
7. Depends on Memory Transaction Type (Table 43); valid if non-zero.
8. x4 Chipkill ECC is always enabled in ganged mode. See F3x44[ChipKillEccEn] for information on symbol size.
9. Retries initiated by either side of the link are logged.
10.

F3x4C MCA NB Status High Register

Cold reset: xxxx xxxxh.
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 also MSRC001_0015[McStatusWrEn].

Bits	Description
31	**Val: error valid**. Read-write; set-by-hardware. 1=This bit indicates that a valid error has been detected. This bit should be cleared to 0 by software after the register has been read.
30	**Over: error overflow**. Read-write; set-by-hardware. 1=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 2.13.1.2.2 [Machine Check Error Logging Overwrite During Overflow] on page 117.
29	**UC: error uncorrected**. Read-write; set-by-hardware. 1=The error was not corrected by hardware.
28	**En: error enable**. Read-write; set-by-hardware. 1=The MCA error reporting is enabled for this error in the MCA Control Register.
27	**MiscV: miscellaneous error register valid**. 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	**AddrV: error address valid**. Read-write; set-by-hardware. 1=The address saved in the address register is the address where the error occurred.
25	**PCC: processor context corrupt**. Read-write; set-by-hardware. 1=The state of the processor may be corrupted by the error condition. Reliable restarting might not be possible.
24:23	Reserved.
22:15	**Syndrome[7:0]: syndrome bits [7:0] for ECC**. Read-write. Logs the lower eight syndrome bits when an ECC error is detected.
14	**CECC: correctable ECC error**. Read-write; set-by-hardware. 1=The error was a correctable ECC error.
13	**UECC: uncorrectable ECC error**. Read-write; set-by-hardware. 1=The error was an uncorrectable ECC error.
12	Reserved.
11:10	**McaStatSubCache: L3 subcache in error**. Indicates the number of the L3 subcache associated with the error. This field is only valid when an L3 error is recorded.
9	**SubLink: sublink or DRAM channel**. 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. DRAM Channel — Sublink 0 = Channel A — 0 = Bits [7:0] 1 = Channel B — 1 = Bits [15:8]
8	**Scrub: error found by DRAM scrubber**. Read-write; set-by-hardware. 1=The error was found by the DRAM scrubber.
7:4	**LDTLink[3:0]**. 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	**ErrCPU[3:0]: error associated with CPU N**. Read-write; set-by-hardware. This field indicates which core within the node is associated with the error. 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.

F3x50 MCA NB Address Low Register

Cold reset: xxxx xxxxh. 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	**ErrAddr[31:1]: Error Address Bits[31:1]**. 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 andare formatted according to Table 50.
- NB array errors indicate the array in error, and are formatted according to Table 51.
- L3 array errors store the physical address which caused the error, and are formatted according to Table 52.
- NB watchdog timer errors are formatted according to Table 53.
- All other NB errors which indicate F3x4C[AddrV] are formatted according to Table 49.

Table 49: Default MCA NB Address Register default encoding

ErrAddr bits	Description
47:1	Physical address bits 47:1
0	Reserved

Table 50: 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.
5:1	0_0000	Link: SRQ Read Response without matching request
	0_0001	Link: Probe Response without matching request
	0_0010	Link: TgtDone without matching request
	0_0011	Link: TgtStart without matching request
	0_0100	Link: Command buffer overflow
	0_0101	Link: Data buffer overflow
	0_0110	Link: Link retry packet count acknowledge overflow
	0_0111	Data command in the middle of a data transfer
	0_1000	Link: Link address extension command followed by a packet other than a command with address.
	0_1001	Link: A specific coherent-only packet from a CPU was issued to an IO link.
	0_1010	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_1011	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_0000	L3: Request gets multiple hits in L3
	1_0001	L3: Probe access gets multiple hits in L3
	1_0010	L3: Request queue overflow
	1_0011	L3: WrVicBlk hit incompatible L3 state
	1_0100	L3: ClVicBlk hit incompatible L3 state
0	-	Reserved

Table 51: MCA NB Address Low Register encoding for NB Array Errors

ErrAddr bits	Array Code	Description
39:6	-	Reserved
5:1	0_0000	SRA: System request address
	0_0001	SRD: System request data
	0_0010	SPB: System packet buffer
	0_0011	MCD: Memory controller data
	0_0100	MPB: Memory packet buffer
	0_0101	LPB0: Link 0 packet buffer
	0_0110	LPB1: Link 1 packet buffer
	0_0111	LPB2: Link 2 packet buffer
	0_1000	LPB3: Link 3 packet buffer
	0_1001	MPBC: Memory controller command packet buffer
	0_1010	MCDBM: Memory controller byte mask
	0_1011	MCACAM: Memory controller address array
	0_1100	DMAP: Extended DRAM address map
	0_1101	MMAP: Extended MMIO address map
	0_1110	X86MAP: Extended PCI/IO address map
	0_1111	CFGMAP: Extended config address map
	1_0000	LPS0: Link 0 packet state buffer
	1_0001	LPS1: Link 1 packet state buffer
	1_0010	LPS2: Link 2 packet state buffer
	1_0011	LPS3: Link 3 packet state buffer
	1_0100	RHB0: Link 0 retry history buffer
	1_0101	RHB1: Link 1 retry history buffer
	1_0110	RHB2: Link 2 retry history buffer
	1_0111	RHB3: Link 3 retry history buffer
	1_1000	SRIMCTRTE: SRI/MCT extended routing table
	1_1001	LN0LN1RTE: Link 0/1 extended routing table
	1_1010	LN2LN3RTE: Link 2/3 extended routing table
	1_1011	GART: GART array
	1_1100	DEV: DEV array
0	-	Reserved

Table 52: MCA NB Address Register encoding for L3 Array Errors

Error Type	Memory Transaction Type (RRRR; Table 43)	ErrAddr bits[1]	Description
Data Error	RD, Evict	47:2	Physical address
	Snoop	47:6	Cache line address requested
		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	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	Cache index
1. The physical address includes the cache index in bits 16:6.			

Table 53: MCA NB Address Low Register encoding for Watchdog Timer Errors

ErrAddr bits	Description
39:36	**System Response Count**. This field records unspecified, implementation-specific information.
35:31	**Wait Code** records unspecified, implementation-specific information (all zeroes means no waiting condition).
30	**Wait for Posted Write**.
29:27	**Destination Node** records the Node ID of the node addressed by the transaction.
26:25	**Destination Unit** is one of: 00 = CPU 01 = GART or DEV Table Walker 10 = Memory Controller 11 = Host
24:22	**Source Node** records the Node ID of the node originating the transaction.
21:20	**Source Unit** (same encoding as Destination Unit)
19:15	**Source Pointer**. 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)
14:11	**SRQ Entry State** records unspecified, implementation-specific information (all zeroes means idle).

Table 53: MCA NB Address Low Register encoding for Watchdog Timer Errors

ErrAddr bits	Description
10:7	**Op Type** records unspecified, implementation-specific information (all zeroes means normal).
6:1	**Link Command**. 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 *HyperTransport™ I/O Link Specification*.

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	**ErrAddr[47:32]: Error Address Bits[47:32]**. Read-write. See F3x50[ErrAddr] for details.

F3x58 Scrub Rate Control Register

Reset: 0000 0000h. This register specifies the ECC scrubbing rate for memory blocks. See also section 2.6.6 [Memory Scrubbers] on page 58. The scrub rate is specified as the time between successive scrub events. A scrub event occurs when a line of memory is checked for errors; the amount of memory that is checked varies based on the memory block (see field descriptions). 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	09h	10.2 us	11h	2.62 ms
02h	80 ns	0Ah	20.5 us	12h	5.24 ms
03h	160 ns	0Bh	41.0 us	13h	10.49 ms
04h	320 ns	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.	

The desired scrubbing rate may vary with different customers; see section 2.13.1.5 [Scrub Rate Recommendations] on page 119 for considerations in setting the scrub rates.

The time required to fully scrub the memory of a node is determined as:
Time = ((memory size in bytes)/64) * (Scrub Rate).
For example, if a node contains 4GB of system memory and DramScrub=1.31ms, then all of the system memory of the node is scrubbed about once every 23 hours.

Bits	Description
31:29	Reserved.
28:24	**L3Scrub: L3 cache scrub rate**. Read-write. Read-write. Specifies time between 64-byte scrub events. Note: the L3 scrubber should be disabled (L3Scrub=00h) if F3xE8[L3Capable]=0 .

23:21	Reserved.
20:16	**DcacheScrub: data cache scrub rate**. 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	**L2Scrub: L2 cache scrub rate**. 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	**DramScrub: DRAM scrub rate**. Read-write. Specifies time between 64-byte scrub events. See also F3x5C and F3x60.

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 also [The MCA NB Control Register] F3x40).

Bits	Description
31:6	**ScrubAddrLo: DRAM scrubber address bits[31:6]**. Read; write to initialize; updated by hardware. Reset: X. 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	**ScrubReDirEn: DRAM scrubber redirect enable**. 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 F3x58[DramScrub]). 1=Enables the scrubber to immediately scrub any address in which a correctable error is discovered. This bit and sequential scrubbing can be enabled independently or together; if both are enabled, the scrubber jumps from the scrubber address to where the correctable error was discovered, scrubs that location, and then jumps back to where it left off; the scrubber address register is not affected during scrubber redirection.

F3x60 DRAM Scrub Address High Register

Reset: see field definitions.

Bits	Description
31:16	Reserved.
15:0	**ScrubAddrHi: DRAM scrubber address bits[47:32]**. Read; write to initialize; updated by hardware. Reset: X. See F3x5C[ScrubAddrLo].

F3x64 Hardware Thermal Control (HTC) Register

See section 2.10.3.1 [PROCHOT_L and Hardware Thermal Control (HTC)] on page 112 for information on HTC. F3x64 is not accessible if [The Northbridge Capabilities Register] F3xE8[HTC capable]=0.

Bits	Description
31	Reserved.
30:28	**HtcPstateLimit: HTC P-state limit select**. Read-write. Reset state varies by product. Specifies the P-state limit of all cores when in the HTC-active state. This field is not changed on a write if the value written is greater than MSRC001_0061[PstateMaxVal]. See also section 2.10.3.1 [PROCHOT_L and Hardware Thermal Control (HTC)] on page 112.
27:24	**HtcHystLmt: HTC hysteresis**. 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 * HtcHystLmt, ranging from 0.0 Tctl to 7.5 Tctl.
23	**HtcSlewSel: HTC slew-controlled temperature select**. Read-write. Reset: 0. 1=HTC logic is driven by the slew-controlled temperature, Tctl, specified in [The Reported Temperature Control Register] F3xA4. 0=HTC logic is driven by the measured control temperature with no slew controls.
22:16	**HtcTmpLmt: HTC temperature limit**. 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 * HtcTmpLmt), ranging from 52.0 Tctl to 115.5 Tctl.
15:8	Reserved.
7	**PslApicLoEn: P-state limit lower value change APIC interrupt enable**. Read-write. Reset: 0. PslApicLoEn and PslApicHiEn enable interrupts using [The Thermal Local Vector Table Entry] APIC330 of each core when the active P-state limit in [The P-State Current Limit Register] MSRC001_0061[CurPstateLimit] changes. PslApicLoEn enables the interrupt when the limit value becomes lower (indicating higher performance). PslApicHiEn enables the interrupt when the limit value becomes higher (indicating lower performance). 1=Enable interrupt.
6	**PslApicHiEn: P-state limit higher value change APIC interrupt enable**. Read-write. Reset: 0. See PslApicLoEn above.
5	**HtcActSts: HTC-active status**. Read; set-by-hardware; write-1-to-clear. Reset: 0. This bit is set by hardware when the processor enters the HTC-active state. It is cleared by writing a 1 to it.
4	**HtcAct: HTC-active state**. Read-only. Reset: X. 1=The processor is currently in the HTC-active state. 0=The processor is not in the HTC-active state.
3:1	Reserved.
0	**HtcEn: HTC enable**. Read-write. Reset: 0. 1=HTC is enabled; the processor is capable of entering the HTC-active state.

F3x68 Software Thermal Control (STC) Register

Reset: ?000 0000h. See section 2.10.3.2 [Software Thermal Control (STC)] on page 112 for information on STC. F3x68 is not accessible if [The Northbridge Capabilities Register] F3xE8[HTC capable]=0.

Bits	Description
31	Reserved.
30:28	**StcPstateLimit: STC P-state limit select**. Read-write. Reset state varies by product. Specifies the P-state limit of all cores when in the STC-active state. This field is not changed on a write if the value written is greater than MSRC001_0061[PstateMaxVal]. See also section 2.10.3.2 [Software Thermal Control (STC)] on page 112.
27:24	**StcHystLmt: STC hysteresis**. Read-write. The processor exits the STC thermal zone when Tctl drops to StcTmpLmt minus StcHystLmt. The encoding is 0.5 * StcHystLmt, ranging from 0.0 Tctl to 7.5 Tctl.
23	**StcSlewSel: STC slew-controlled temperature select**. Read-write. Reset: 0. 1=STC logic is driven by the slew-controlled temperature, Tctl, specified in [The Reported Temperature Control Register] F3xA4. 0=STC logic is driven by the measured control temperature with no slew controls.
22:16	**StcTmpLmt: STC temperature limit**. 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 * StcTmpLmt), ranging from 52.0 Tctl to 115.5 Tctl.
15	Reserved.
14:12	Reserved.
11	Reserved.
10:8	Reserved.
7	**StcTmpLoSts: STC temperature low status**. Read; write-1-to-clear. This bit is set high when the processor exits the STC thermal zone.
6	**StcTmpHiSts: STC temperature high status**. Read; write-1-to-clear. This bit is set high when the processor enters the STC thermal zone.
5	**StcPstateEn: STC P-state enable**. Read-write. 1=Place the processor into the STC-active state.
4	Reserved.
3	**StcApcTmpLoEn: STC APIC temperature low interrupt enable**. 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.
2	**StcApcTmpHiEn: STC APIC temperature high interrupt enable**. Read-write. 1=Enables the generation of an interrupt using [The Thermal Local Vector Table Entry] APIC330 of each core when the processer enters the STC thermal zone.
1	**StcSbcTmpLoEn: STC special bus cycle temperature low enable**. 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.
0	**StcSbcTmpHiEn: STC special bus cycle temperature high enable**. 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.

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 >= 1 DnReqDBC >= 1 UpReqDBC >= 1
  DnRspDBC >=1

- If F0x68[DispRefModeEn] is set or any of the F0x[E4, C4, A4, 84][IsocEn] bits are set: IsocRspDBC >= 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] <= 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] <= 24
- If the system is a UMA system, DatBuf24 should be set and the following buffer allocations should be used:
  UpRspDBC = 1 DnReqDBC = 1 UpReqDBC = 1
  DnRspDBC =1 IsocRspDBC = 6
- If the system is not a UMA system, DatBuf24 should be set and the following buffer allocations should be used:
  UpRspDBC = 1 DnReqDBC = 1 UpReqDBC = 2
  DnRspDBC =1 IsocRspDBC = 0[1]/1[2]

1. If all of the F0x[E4, C4, A4, 84][IsocEn] bits are clear.
2. If any of the F0x[E4, C4, A4, 84][IsocEn] bits are set.

Bits	Description
31	Reserved.
30:28	**IsocRspDBC: isochronous response data buffer count**. Read-write. Cold reset: 3.
27:19	Reserved.
18:16	**UpRspDBC: upstream response data buffer count**. Read-write. Cold reset: 2.
15	**DatBuf24: data buffer allocation 24**. 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	**DnRspDBC: downstream response data buffer count**. Read-write. Cold reset: 2.
5:4	**DnReqDBC: downstream request data buffer count**. Read-write. Cold reset: 1.
3	Reserved.
2:0	**UpReqDBC: upstream request data buffer count**. 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:
  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:
  IsocRspCBC + IsocPreqCBC + IsocReqCBC + UpRspCBC + DnPreqCBC + UpPreqCBC + DnReqCBC +

DnRspCBC + UpReqCBC + F3x7C[Sri2XbarFreeRspCBC] + F3x7C[Sri2XbarFreeXreqCBC] <= 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:
  IsocRspCBC + IsocPreqCBC + IsocReqCBC + UpRspCBC + DnPreqCBC + UpPreqCBC + DnReqCBC + DnRspCBC + UpReqCBC + F3x7C[Sri2XbarFreeRspCBC] + F3x7C[Sri2XbarFreeXreqCBC] <= 32
- If the system is a UMA system, F3x6C[DatBuf24] should be set and the following buffer allocations should be used:

UpRspCBC = 2	UpPreqCBC = 1	DnPreqCBC = 1
UpReqCBC = 1	DnReqCBC = 1	DnRspCBC = 1
IsocReqCBC = 2	IsocRspCBC = 6	IsocPreqCBC = 1

- If the system is not a UMA system, F3x6C[DatBuf24] should be set and the following buffer allocations should be used:

UpRspCBC = 4	UpPreqCBC = 1	DnPreqCBC = 1
UpReqCBC = 3	DnReqCBC = 1	DnRspCBC = 1
IsocReqCBC = $0^1/1^2$	IsocRspCBC = $0^1/1^2$	IsocPreqCBC = $0^1/1^2$

1. If all of the F0x[E4, C4, A4, 84][IsocEn] bits are clear.
2. If any of the F0x[E4, C4, A4, 84][IsocEn] bits are set.

Bits	Description
31	Reserved.
30:28	**IsocRspCBC: isochronous response command buffer count**. Read-write. Cold reset: 6.
27	Reserved.
26:24	**IsocPreqCBC: isochronous posted request command buffer count**. Read-write. Cold reset: 1.
23	Reserved.
22:20	**IsocReqCBC: isochronous request command buffer count**. Read-write. Cold reset: 7.
19	Reserved.
18:16	**UpRspCBC: upstream response command buffer count**. Read-write. Cold reset: 4.
15	Reserved.
14:12	**DnPreqCBC: downstream posted request command buffer count**. Read-write. Cold reset: 4.
11	Reserved.
10:8	**UpPreqCBC: upstream posted request command buffer count**. Read-write. Cold reset: 4.
7:6	**DnRspCBC: downstream response command buffer count**. Read-write. Cold reset: 2.
5:4	**DnReqCBC: downstream request command buffer count**. Read-write. Cold reset: 2.
3	Reserved.
2:0	**UpReqCBC: upstream request command buffer count**. 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 multiprocessor system the following minimum buffer allocations are required:

ProbeCBC >= 2	DnPreqCBC >= 1	UpPreqCBC >= 1
DnReqCBC >= 1	UpReqCBC >= 1	

- To ensure deadlock free operation in a uniprocessor system the following minimum buffer allocations are required:

ProbeCBC >= 2 UpReqCBC >= 1 UpPreqCBC >= 1

- To ensure deadlock free operation ProbeCBC must be less than or equal to 8.
- If F0x68[DispRefModeEn] is set:
  IsocReqCBC >= 1
- If any of the F0x[E4, C4, A4, 84][IsocEn] bits are set:
  IsocPreqCBC >= 1 IsocReqCBC >= 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 >= 1 IsocReqCBC >= 1 DRReqCBC >=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:
  DRReqCBC + IsocPreqCBC + IsocReqCBC + DnPreqCBC + UpPreqCBC + DnReqCBC + UpReqCBC + F3x7C[Xbar2SriFreeListCBC] + F3x1A0[L3ToSriReqCBC] <= 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:
  DRReqCBC + IsocPreqCBC + IsocReqCBC + DnPreqCBC + UpPreqCBC + DnReqCBC + UpReqCBC + F3x7C[Xbar2SriFreeListCBC] + (F3x1A0[CpuCmdBufCnt] * (F3xE8[CmpCap] +1)) <= 32
- If the system is a UMA system and F2x118[MctPriIsoc]=11b, the following settings should be used:
  DRReqCBC = 12 IsocPreqCBC = 1 IsocReqCBC = 1
  ProbeCBC = 8 UpPreqCBC = 1 DnReqCBC = 0
  DnPreqCBC = 0 UpReqCBC = 1 F3x7C[Xbar2SriFreeListCBC] = 12
  F2x118[MctVarPriCntLmt] = 1
- If the system is a UMA system and F2x118[MctPriIsoc]=10b, the following settings should be used:
  DRReqCBC = 9 IsocPreqCBC = 1 IsocReqCBC = 1
  ProbeCBC = 8 UpPreqCBC = 1 DnReqCBC = 0
  DnPreqCBC = 0 UpReqCBC = 1 F3x7C[Xbar2SriFreeListCBC] = 15
- If the system is a UMA system and 32 byte display refresh requests are generated by the graphics engine, the following settings should be used:
  DRReqCBC = 15 IsocPreqCBC = 1 IsocReqCBC = 1
  ProbeCBC = 8 UpPreqCBC = 1 DnReqCBC = 0
  DnPreqCBC = 0 UpReqCBC = 1 F3x7CXbar2SriFreeListCBC = 9

Bits	Description
31:28	**DRReqCBC: display refresh request command buffer count**. Read-write. Cold reset: 0.
27	Reserved.
26:24	**IsocPreqCBC: isochronous posted request command buffer count**. Read-write. Cold reset: 0.
23:20	**IsocReqCBC: isochronous request command buffer count**. Read-write. Cold reset: 4.
19:16	**ProbeCBC: probe command buffer count**. Read-write. Cold reset: 8.
15	Reserved.
14:12	**DnPreqCBC: downstream posted request command buffer count**. Read-write. Cold reset: 1.
11	Reserved.
10:8	**UpPreqCBC: upstream posted request command buffer count**. Read-write. Cold reset: 1.
7	Reserved.
6:4	**DnReqCBC: downstream request command buffer count**. Read-write. Cold reset: 1.
3	Reserved.
2:0	**UpReqCBC: upstream request command buffer count**. 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:
  ProbeCBC >= 1 RspCBC >= 1 RspDBC >= 2
  RspDBC >= F2x11C[MctPrefReqLimit]+1
- To ensure deadlock free operation when online spare is enabled (F2x[1, 0][5C:40][Spare] = 1) the following minimum buffer allocation is required:
  RspCBC >= D
- The total number of command buffers allocated in this register must satisfy the following equation:
  ProbeCBC + RspCBC <= 32

Bits	Description
31:22	Reserved.
21:16	**RspDBC: response data buffer count**. 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	**ProbeCBC: probe command buffer count**. Read-write. Cold reset: Ch.
7:5	Reserved.
4:0	**RspCBC: response command buffer count**. 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 Sri2XbarFreeRspCBC = 0: Sri2XbarFreeXreqCBC >2
  - If Sri2XbarFreeRspCBC != 0: Sri2XbarFreeRspCBC > 2
  - If Sri2XbarFreeRspDBC = 0: Sri2XbarFreeXreqDBC >2
  - If Sri2XbarFreeRspDBC != 0: Sri2XbarFreeRspDBC > 2
  - Xbar2SriFreeListCBC >= 2
- If the system is a UMA system, the following buffer allocations should be used:
  Sri2XbarFreeXreqCBC = 8 Sri2XbarFreeXreqDBC = 7
  Sri2XbarFreeRspCBC = 0 Sri2XbarFreeRspDBC = 0
- If the system is not a UMA system, the following buffer allocations should be used:
  Sri2XbarFreeXreqCBC = 9 Sri2XbarFreeXreqDBC = 9
  Sri2XbarFreeRspCBC = 0 Sri2XbarFreeRspDBC = 0

Bits	Description
31	Reserved.
30:28	**Xbar2SriFreeListCBInc: XBAR to SRI free list command buffer increment**. Read-write. Cold reset: 0. This is use to add buffers to the free list pool if they are reclaimed from hard allocated entries without having to go through warm reset.
27:23	Reserved.
22:20	**Sri2XbarFreeRspDBC: SRI to XBAR free response data buffer count**. Read-write. Cold reset: 3.

19:16	**Sri2XbarFreeXreqDBC: SRI to XBAR free request and posted request data buffer count**. Read-write. Cold reset: 3. When Sri2XbarFreeRspDBC=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.
15:12	**Sri2XbarFreeRspCBC: SRI to XBAR free response command buffer count**. Read-write. Cold reset: 15.
11:8	**Sri2XbarFreeXreqCBC: SRI to XBAR free request and posted request command buffer count**. 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	**Xbar2SriFreeListCBC: XBAR to SRI free list command buffer count**. Read-write. Cold reset: varies based on the state of F3xE8[CmpCap] and whether the processor includes L3 cache: • 1-core without L3cache is 22. • 2-core or any processor with L3 cache is 20. • 3-core without L3cache is 18. • 4-core without L3cache is 16.

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] on page 27 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 54: 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
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. C1Eis not supported in multi-link or multi-socket systems.	F3x80[15:8]: A6h
FIDVID change	010b	Unused	F3x80[23:16]: 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. Refer to section 1.5.2 [Supported Feature Variations] on page 21 for package-specific support. AMD recommends using PROCHOT_L for thermal throttling and not implementing stop clock based throttling.	F3x84[15:8]: 41h

Table 54: ACPI Power State Control Register SMAF Settings

S4/S5	110b	Initiated by a processor access to the ACPI-defined PM1_CNTa register.	F3x84[23:16]:E6h
C1	111b	Initiated when a Halt instruction is executed by processor. This does not involve the interaction with the SMC, therefore the SMC is required to never send STPCLK assertion commands with SMAF=7h.	F3x84[31:24]: With L2 and data cache scrubbing disabled: A0h With L2 or data cache scrubbing enabled: 80h. See section 2.6.6 [Memory Scrubbers] on page 58.
1. See section 2.6.4.2.4 [F0x[5C:40]Display Refresh And IFCM] on page 57.			

Bits	Description
31:8	See above.
7:5	**ClkDivisor: clock divisor**. Read-write. Specifies the core clock frequency while in the low-power state. This divisor is relative to the current FID frequency, or: • 100 MHz * (10h + MSRC001_00[68:64][CpuFid]) of the current P-state specified by MSRC001_0063[CurPstate]. 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: Bits Divisor Bits Divisor 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. See also section 2.6.6 [Memory Scrubbers] on page 58.
4:3	Reserved.
2	**NbGateEn: Northbridge gate enable**. Read-write. 1=The NB clock is mostly gated off and MEMCLK is tristated when LDTSTOP_L is asserted while in the low-power state. Setting this bit further reduces dynamic power while in the low-power state. NbLowPwrEn is required to be set if this bit is set.
1	**NbLowPwrEn: Northbridge low-power enable**. Read-write. 1=The NB clock is ramped down to the divisor specified by [The Clock Power/Timing Control 0 Register] F3xD4[NbClkDiv] and DRAM is placed into self-refresh mode when LDTSTOP_L is asserted while in the low-power state.
0	**CpuPrbEn: CPU direct probe enable**. Read-write. Specifies how probes are handled while in the low-power state. 0=When the probe request comes into the NB, the core clock is brought up to the COF (based on the current P-state), all outstanding probes are completed, the core waits for a hysteresis time based on [The Clock Power/Timing Control 0 Register] F3xD4[ClkRampHystSel], and then the core clock is brought down to the frequency specified by ClkDivisor. 1=The core clock does not change frequency; the probe is handled at the frequency specified by ClkDivisor; this may only be set if: • ClkDivisor specifies a divide-by 1, 2, 4, 8, or 16 and NbCof <= 3.2 GHz • ClkDivisor specifies a divide-by 1, 2, 4, or 8 and NbCof >= 3.4 GHz 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	**DisGartTblWlkPrb: disable GART table walk probes**. 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	**DisGartIo: disable GART IO accesses**. Read-write. 1=Disables requests from IO devices from accessing the GART.
4	**DisGartCpu: disable GART CPU accesses**. Read-write. 1=Disables requests from CPUs from accessing the GART.
3:1	**GartSize: GART size**. 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	**GartEn: GART enable**. Read-write. 1=Enables GART address translation for accesses falling within the GART aperture. F3x94[GartAperBaseAddr] and other related registers should be initialized before GartEn is set.

F3x94 GART Aperture Base Register

Reset: See field definitions.

Bits	Description
31:15	Reserved.
14:0	**GartAperBaseAddr[39:25]: GART aperture base address bits[39:25]**. Read-write. Reset: X. Specifies the base address of the GART aperture range. Based on F3x90[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 F3x90[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	**GartTblBaseAddr[39:12]: GART table base address bits[39:12]**. 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 F3x90 and F3x94, 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: 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. 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.
3:0	Reserved.

F3x9C GART Cache Control Register

Reset: 0000 0000h.

Bits	Description
31:2	Reserved.
1	**GartPteErr: GART PTE error**. Read; set-by-hardware; write-1-to-clear. 1=An invalid PTE was encountered during a table walk.
0	**InvGart: invalidate GART**. 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.

Bits	Description
31	**CofVidProg: COF and VID of P-states programmed**. Read-only. 1=Out of cold reset, the VID and FID values of the P-state register specified by MSRC001_0071[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 F3xD4[NbFidEn]. Registers containing P-state information such as FID, DID, and VID values are valid out of cold reset independent of the state of F3xA0[CofVidProg]. BIOS must transition the processor to a valid P-state out of cold reset when F3xA0[CofVidProg]=0. See also section 2.4.2.5 [BIOS Requirements for P-State Initialization and Transitions] on page 37.
30	Reserved.

29	**SlamVidMode: slam voltage ID mode**. 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 also section 2.4.1.7 [Hardware-Initiated Voltage Transitions] on page 32. BIOS should set this bit to the inverse of F3xA0[PviMode].
28	Reserved.
27:16	**PstateId: P-state identifier**. Read-only. This field specifies the P-state ID associated with the product.
15:14	Reserved.
13:11	**PllLockTime: PLL synchronization lock time**. 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: 000b 1 microsecond. 100b 8 microseconds. 001b 2 microseconds. 101b 16 microseconds. 010b 3 microseconds. 110b Reserved. 011b 4 microseconds. 111b Reserved. BIOS should set this field to 101b.
10:9	Reserved.
8	**PviMode: parallel VID interface mode**. Read-only. 1=The parallel VID interface is selected (through a resistor strap on VID[1] to VDDIO); single- or dual-plane operation. 0=The serial VID interface is selected (through a resistor strap on VID[1] to ground); dual-plane operation. See section. 2.4.1.1 [VID Pins And Interface Selection] on page 29.
7	**PsiVidEn: PSI_L VID enable**. 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 section 2.4.1.4 [PSI_L] on page 30.
6:0	**PsiVid: PSI_L VID threshold**. 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 section 2.4.1.4 [PSI_L] on page 30.

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	**CurTmp: current temperature**. 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 also section 2.10.1 [The Tctl Temperature Scale] on page 111. See also CurTmpSel.
20:18	Reserved.
17:16	**CurTmpSel**. 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	**PerStepTimeDn[4:0]: per 1/8th degree step time down**. 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.
7	**TmpSlewDnEn: temperature slew downward enable**. 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	**TmpMaxDiffUp: temperature maximum difference up**. 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.
4:0	**PerStepTimeUp[4:0]: per 1/8th degree step time up**. 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: 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.

F3xB0 On-Line Spare Control Register

See 2.8.11 [On-Line Spare] on page 107 for more details on on-line spare.

Bits	Description
31:28	**LvtOffset: local vector table offset**. 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	**EccErrCnt: ECC error count**. 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	**EccErrCntWrEn: ECC error counter write enable**. Read-write. Reset: 0. 1=Enable writes to the EccErrCnt field.
22	Reserved.
21:20	**EccErrCntChan: ECC error counter channel**. Read-write. Reset: 0. These bits specify the channel for which ECC error count information is returned in the EccErrCnt field as follows: Memory Interface / Bits[21:20] / Channel DDR / x0 / DCT 0 (channel A) DDR / x1 / DCT 1 (channel B)
19:16	**EccErrCntDramCs[3:0]: ECC error counter DRAM chip select**. 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: • 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].
15:14	**EccErrInt: ECC error interrupt type**. 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. 00b = No Interrupt. 01b = APIC based interrupt (see LvtOffset) to all cores. 10b = SMI trigger event (always routed to CpuCoreNum 0, as defined in section 2.9.2 [CPU Cores and Downcoring] on page 108); see section 2.14.2.3 [SMI Sources And Delivery] on page 128. 11b = Reserved.
13:12	**SwapDoneInt: swap done interrupt type**. Read-write. Reset: 0. This field specifies the type of interrupt generated when a swap is complete. 00b = No Interrupt. 01b = APIC based interrupt (see LvtOffset) to all cores. 10b = SMI trigger event (always routed to CpuCoreNum 0, as defined in section 2.9.2 [CPU Cores and Downcoring] on page 108); see section 2.14.2.3 [SMI Sources And Delivery] on page 128. 11b = Reserved.
11	Reserved.
10:8	**BadDramCs1: bad DRAM chip select DCT1**. Read-write. Cold reset: 0. See BadDramCs0 below.
7	Reserved.
6:4	**BadDramCs0: bad DRAM chip select DCT0**. 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.
3	**SwapDone1: swap done DCT1**. Read-write; set-by-hardware. Cold reset: 0. See SwapDone0 below.
2	**SwapEn1: swap enable DCT1**. Read; write-1-only. Cold reset: 0. See SwapEn0 below.

1	**SwapDone0: swap done DCT0**. 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	**SwapEn0: swap enable DCT0**. Read; write-1-only. Cold 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.

F3xD4 Clock Power/Timing Control 0 Register

Reset: see field definitions.

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>31</td><td>NbClkDivApplyAll. Read-write. Cold reset: 0b. See NbClkDiv. BIOS should set this bit to 1b.</td></tr>
<tr><td>30:28</td><td>NbClkDiv: NB clock divisor. Read-write. Cold reset: value varies by product. Specifies the NB CLK divisor associated with [The ACPI Power State Control Registers] F3x[84:80][NbLowPwrEn]. This divisor is applied while LDTSTOP is asserted if the corresponding core CLK divisor, F3x[84:80][ClkDivisor], is set to “turn off clocks” or if NBClkDivApplyAll=1; otherwise, the divisor specified by F3x[84:80][ClkDivisor] is applied. This divisor is relative to the current NB FID frequency, or:
• 200 MHz * (4 + F3xD4[NbFid]).
If MSRC001_00[68:64][NbDid] 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:
<table>
<tr><th>Bits</th><th>Divisor</th><th>Bits</th><th>Divisor</th></tr>
<tr><td>000b</td><td>Divide-by 1.</td><td>100b</td><td>Divide-by 16.</td></tr>
<tr><td>001b</td><td>Divide-by 2.</td><td>101b</td><td>Reserved.</td></tr>
<tr><td>010b</td><td>Divide-by 4.</td><td>110b</td><td>Reserved.</td></tr>
<tr><td>011b</td><td>Divide-by 8.</td><td>111b</td><td>Reserved.</td></tr>
</table>
BIOS should set this field to 100b.</td></tr>
</table>

<table>
<tr><td>27:24</td><td>PowerStepUp. 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 15 steps in this transition of each core and about 5 steps for the NB for the PowerStepDown and PowerStepUp transitions. So the total transition time for a single core is about 15 times the time specified by PowerStepDown and PowerStepUp and the transition time for the NB is about 5 times the time specified by PowerStepDown and PowerStepUp. Use of longer transition times may help reduce voltage transients associated with power state transitions. The bits for PowerStepUp and PowerStepDown are encoded as follows:
<table>
<tr><th>Bits</th><th>Time</th><th>Bits</th><th>Time</th><th>Bits</th><th>Time</th><th>Bits</th><th>Time</th></tr>
<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>
</table>
Note that if PowerStepDown or PowerStepUp are programmed to greater than 50ns, then the value applied to NB steps is clipped to 50ns. BIOS should set the PowerStepDown and PowerStepUp fields to 1000b for all processors in desktop and mobile systems, and use the following equation for processors in server systems: PowerStepDown (ns) = PowerStepUp (ns) = 400 / # of cores (e.g. 4 cores = 400 / 4 = 100ns = 0011b).</td></tr>
<tr><td>23:20</td><td>PowerStepDown. 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.</td></tr>
<tr><td>19:18</td><td>Reserved.</td></tr>
<tr><td>17:16</td><td>LnkPllLock. 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.
<table>
<tr><th>Bits</th><th>PLL lock time</th></tr>
<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>
</table></td></tr>
<tr><td>15:12</td><td>Reserved.</td></tr>
<tr><td>11:8</td><td>ClkRampHystSel: clock ramp hysteresis select. Read-write. Cold reset: 000b. 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.

If F3x[84:80][CpuPrbEn]=0 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. The hysteresis time is encoded as 320ns * (1 + ClkRampHystSel). If F3x[84:80][CpuPrbEn]=1 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 seviced 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 40ns * (1 + ClkRampHystSel).
BIOS should set this field to 1111b.
</td></tr>
<tr><td>7:6</td><td>Reserved.</td></tr>
</table>

5	**NbFidEn: Northbridge frequency ID enable**. Read-write. Cold reset: value is the same as F3xA0[CofVidProg]. 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, regardess of the state of NbFid.1=The NB FID is specified by NbFid. See also section 2.4.2 [P-states] on page 34.
4:0	**NbFid: Northbridge frequency ID**. 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: • NB COF = 200 MHz * (F3xD4[NbFid] + 4h) / (2^MSRC001_00[68:64][NbDid]). This field must be programmed to the requirements specified in MSRC001_0071[MaxNbFid] 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 2.4.2.6 [BIOS Northbridge COF and VID Configuration] on page 37. See also section 2.4.2 [P-states] on page 34. 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	**ReConDel: link reconnect delay**. 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 F0x[E4, C4, A4, 84][LdtStopTriEn]=1 and F0x[18C:170][LS2En]=1. The assertion of CTL is delayed until the specified time has elapsed. See section 2.7.6 [Link LDTSTOP_L Disconnect-Reconnect] on page 61 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.
23:15	Reserved.
14:8	**TdpVid: thermal design power VID**. Read-write. Cold reset: value varies by product. This specifies the voltage used to calculate TDP during P0. P0 TDP = TdpVid * ProcIddMax. For all platforms, use the TdpVid encoding specified in Table 6: [SVI and internal VID codes] on page 31 and the single plane ProcIddMax equation defined in section 2.4.2.7 [Processor-Systemboard Power Delivery Compatibility Check] on page 39.
7	Reserved.

6:4	**VSRampTime: voltage stabilization ramp time**. 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 2.4.1.5 [VID Encodings] on page 30 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 also section 2.4.1.7 [Hardware-Initiated Voltage Transitions] on page 32. Bits Time Bits Time 000b 10 microseconds 100b 60 microseconds 001b 20 microseconds 101b 100 microseconds 010b 30 microseconds 110b 200 microseconds 011b 40 microseconds 111b 500 microseconds BIOS should set this field to 001b.
3	Reserved.
2:0	**VSSlamTime: voltage stabilization slam time**. 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 also section 2.4.1.7 [Hardware-Initiated Voltage Transitions] on page 32. If in SVI mode, then this time measures the period after the end of the SVI command. Bits Time Bits Time 000b 10 microseconds 100b 60 microseconds 001b 20 microseconds 101b 100 microseconds 010b 30 microseconds 110b 200 microseconds 011b 40 microseconds 111b 500 microseconds BIOS should program this field according to the following equation: VSSlamTime = .4us/mV * [P0 voltage - Pmin voltage]. The VSSlamTime value should be rounded up to the nearest programmable time.

F3xDC Clock Power/Timing Control 2 Register

Reset: see field definitions.

Bits	Description

31:15	Reserved.
14:12	**NbsynPtrAdj: NB/core synchronization FIFO pointer adjust**. Read-write. Cold reset: 000b. 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. BIOS should program this field to 5h for optimal performance. 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. ... ... 7h Position the read pointer 7 clock cycles closer to the write pointer.
11	Reserved.
10:8	**PstateMaxVal: P-state maximum value**. 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 also MSRC001_0061[PstateMaxVal].
7:0	Reserved.

F3xE4 Thermtrip Status Register

Reset: 0000 0000h, except bits[14:8, 5, 3 and 1]; see below.

Bits	Description
31	**SwThermtp: software THERMTRIP**. 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	**DiodeOffset**. Read-only. Reset: value varies by product. This field is used to specify the correction value applied to thermal diode measurements. See also section 2.10.2 [Thermal Diode] on page 111. It is encoded as follows: 00h is undefined. 01h to 3Fh: correction = +11C - DiodeOffset, or {01h to 3Fh} = {+10C to -52C}. 40h to 7Fh: undefined.
7:6	Reserved.
5	**ThermtpEn: THERMTRIP enable**. Read-only. 1=The THERMTRIP state as specified in section 2.10.3.3 [THERMTRIP] on page 113 is supported by the processor.
4	Reserved.
3	**ThermtpSense: THERMTRIP** sense. Read-only. Cold reset: 0. 1=The processor temperature exceeded the THERMTRIP value (regardless as to whether the THERMTRIP state is enabled).
2	Reserved.

1	**Thermtp: THERMTRIP**. 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
31:26	Reserved.
25	**L3Capable**. 1=Specifies that an L3 cache is present. See also CPUID Fn8000_0006_EDX.
24	Reserved.
23:20	**UnGangEn: link unganging enabled**. 0=Link is forced into the ganged state and may not be placed into the unganged 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 also section 2.7 [Links] on page 59.
19	Reserved.
18:16	**MpCap: MP capability**. 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	Reserved.
14	**Multiple VID plane capable**.
13:12	**CmpCap: CMP capable**. Specifies the number of cores enabled on the node. 00b=1; 01b=2; 10b=3; 11b=4.
11	**LnkRtryCap**. **Link error-retry capable**.
10	**HTC capable**. This affects F3x64 and F3x68.
9	**SVM capable**.
8	**MctCap: memory controller (on the processor) capable**.
7:5	**DdrMaxRate**. Specifies the maximum DRAM data rate that the processor is designed to support. Bits DDR limit Bits DDR limit 000b No limit 100b 800 MT/s 001b 1600 MT/s 101b 667 MT/s 010b 1333 MT/s 110b 533 MT/s 011b 1067 MT/s 111b 400 MT/s
4	**Chipkill ECC capable**.
3	**ECC capable**.
2	**Eight-node multi-processor capable**.
1	**Dual-node multi-processor capable**.
0	**DctDualCap: two-channel DRAM capable (i.e., 128 bit)**. 0=Single channel (64-bit) only.

F3xF0 DEV Capability Header Register

See section 2.6.3 [DMA Exclusion Vectors (DEV)] on page 54. Note: if SVM is not supported, as specified by F3xE8[SVM Capable], then this register is reserved. See also Table 1 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 section 2.6.2 [The GART] on page 54). 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	**IntCap: interrupt reporting capability**. Read-only. Reset: 0. 0=Indicates that interrupt reporting of DEV protection violations is not present on this device.
20	**MceCap: MCE reporting capability**. Read-only. Reset: 1. Indicates that machine check architecture reporting of DEV protection violations is present on this device.
19	Reserved.
18:16	**CapType: DEV capability block type**. Read-only. Reset: 000b. Specifies the layout of the Capability Block.
15:8	**CapPtr: capability pointer**. Read-only. Reset: 00h. Indicates that this is the last capability block.
7:0	**CapId: capability ID**. Read-only. Reset: 0Fh. Indicates a DEV capability block.

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	**DevFunction**. Read-write. See F3xF0 for details. Valid values for this field are 00h through 07h.
7:0	**DevIndex**. 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	**BaseAddress[31:12]: DEV table base address bits[31:12]**. 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. If any part of the DEV table is in other than system memory, then undefined behavior results.
11:7	Reserved.
6:2	**Size: DEV table size**. Read-write. These bits specify the size of the memory region that the DEV table covers, 4GB*(2^Size). The corresponding DEV table size is 128KB*(2^Size).
1	**Protect: protect out-of-range addresses**. Read-write. 0=DMA accesses to addresses that are outside the range covered by the DEV table are allowed. 1=DMA accesses to addresses that are outside the range covered by the DEV table are protected.
0	**Valid: DEV table valid**. Read-write. 1=The DEV table for the protection domain specified by 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	**BaseAddress[47:32]: DEV table base address bits[47:32]**. Read-write. See F3xF8_x0[BaseAddress].

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	**Dom1: protection domain 1**. 3 LSBs are read-write; 3 MSBs are read-only, 000b. This is the protection domain number assigned to Unit1.
25:20	**Dom0: protection domain 0**. 3 LSBs are read-write; 3 MSBs are read-only, 000b. This is the protection domain number assigned to Unit0.
19:12	**BusNu: bus number**. Read-write.
11	**Valid1: UnitID 1 valid**. Read-write. 1=Enable DEV checking for Unit1 and Dom1.
10:6	**Unit1: IO link UnitID 1**. Read-write.
5	**Valid0: UnitID 0 valid**. Read-write. 1=Enable DEV checking for Unit0 and Dom0.
4:0	**Unit0: IO link UnitID 0**. Read-write.

F3xF8_x3 DEV Capabilities Register

Bits	Description
31:24	Reserved.
23:16	**NMaps: number of map registers implemented**. Read-only, 04h. Specifies the number of instantiations of F3xF8_x2.
15:8	**NDomains: number of protection domains implemented**. Read-only, 08h. Specifies the number of protection domains and the number of instantiations of F3xF8_x0 and F3xF8_x1.
7:0	**Revision: DEV register-set revision number**. Read-only, 00h.

F3xF8_x4 DEV Control Register

Reset: 0000 0000h.

Bits	Description
31:7	Reserved.
6	**DevTblWalkPrbDis: DEV table walk probe disable**. Read-write. 1=Disable probing of CPU caches during DEV table walks. This bit may be set to improve DEV cache table walk performance when the DEV is in non-cacheable or write-through memory.
5	**SlDev: secure loader DEV protection enable**. Read-write; 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	**DevInv: invalidate DEV cache**. Read; write-1-only. 1=Invalidate the DEV table-walk cache. This bit is cleared by hardware when invalidation is complete.
3	**MceEn: MCE reporting enable**. Read-write. 1=Enable reporting of DEV protection violations through a machine check exception.
2	**IoDis: upstream IO disable**. Read-write; set-by-hardware. This bit is set by hardware after an SKINIT instruction. 1=Upstream IO-space accesses are regarded as DEV protection violations.
1	Reserved. Read-write.
0	**DevEn: DEV enable**. 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	**ErrVal: error valid**. Read-write; set-by-hardware. 1=A valid DEV protection violation has been logged in this register.
30	**ErrOver: error overflow**. Read-write; set-by-hardware. 1=A DEV protection violation was detected while ErrVal was set for a prior violation. DEV protection violations detected while ErrVal is set are not logged in this register.
29	**ErrAddrVal: error address valid**. Read-write; set-by-hardware. 1=The address saved in F3xF8_x6 and F3xF8_x7 is the address associated with the error.
28:24	Reserved.
23:16	**ModelSpecErr: model specific error**. Read-only, 00h.
15:8	Reserved.
7:5	**ErrCodeDest: error code destination**. Read-write; set-by-hardware. Specifies the destination of the transaction that resulted in the protection violation. 000b = Generic (or could not be determined) 001b = DRAM 010b = MMIO 011b = reserved 100b = IO space 101b = Configuration 110b = reserved 111b = reserved
4:2	**ErrCodeSrc: error code source**. Read-write; set-by-hardware. Specifies the source of the transaction that resulted in the protection violation. 000b = Generic (or could not be determined) 001b = CPU 010b = IO device 011b - 111b = reserved
1:0	**ErrCodeAccType: error code access type**. Read-write; set-by-hardware. Specifies the access type of the transaction that resulted in the protection violation. 00b = Generic (or could not be determined) 01b = Read 10b = Write 11b = Read-modify-write

F3xF8_x6 DEV Error Address Low Register

Cold reset: 0000 0000h.

Bits	Description
31:2	**ErrAddr: error address bits[31:2]**. Read-write; set-by-hardware. See F3xF8_x5 for details.
1:0	Reserved.

F3xF8_x7 DEV Error Address High Register

Cold reset: 0000 0000h.

Bits	Description
31:16	Reserved.
15:0	**ErrAddr: error address bits[47:32]**. Read-write; set-by-hardware. See F3xF8_x5 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	**ExtFamily: extended family**. Read-only.
19:16	**ExtModel: extended model**. Read-only.
15:12	Reserved.
11:8	**BaseFamily**. Read-only.
7:4	**BaseModel**. Read-only.
3:0	**Stepping**. Read-only.

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 also section 2.6.1 [Northbridge (NB) Architecture] on page 54. 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/sublink 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 preferential access to these tokens.
- If an IOMMU is present on a link, F3x1[54, 50, 4C, 48][IsocReqTok] for that link 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.
- If the system is a UMA system using display refresh mode, the following XCS token settings should be used:

FreeTok = 12	IsocRspTok = 3	IsocPreqTok = 1
IsocReqTok = 3	DnRspTok = 1	UpRspTok = 3
DnPreqTok = 1	UpPreqTok = 1	DnReqTok = 1
UpReqTok = 2	F3x144[ProbeTok] = 3	F3x144[RspTok] = 6
F3x148[IsocRspTok0] = 0	F3x148[IsocPreqTok0] = 1	F3x148[IsocReqTok0] = 1

F3x148[ProbeTok0] = 0 F3x148[RspTok0] = 2 F3x148[PReqTok0] = 2
F3x148[ReqTok0] = 2 F3x148[FreeTok] = 8 F3x158[LnkToXcsDRToken] = 3

- If the system is not using IFCM or display refresh mode, the following XCS token settings should be used:

FreeTok = 8 IsocRspTok = 0 IsocPreqTok = 0
IsocReqTok =0 DnRspTok = 1 UpRspTok = 3
DnPreqTok = 1 UpPreqTok = 1 DnReqTok = 1
UpReqTok = 2 F3x144[ProbeTok] = 3 F3x144[RspTok] = 3
For each enabled link:
F3x1[50:48][IsocRspTok0] = 0 F3x1[50:48][IsocPreqTok0] = 0 F3x1[50:48][IsocReqTok0] = 0
F3x1[50:48][ProbeTok0] = 2 F3x1[50:48][RspTok0] = 2 F3x1[50:48][PReqTok0] = 2
F3x1[50:48][ReqTok0] = 2 F3x1[50:48][FreeTok] = 3

Updates to this register do not take effect until after a warm reset.

Bits	Description
31:24	Reserved.
23:20	**FreeTok: free tokens**. 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	**IsocRspTok: isochronous response tokens**. Read-write. Cold reset: 0.
15:14	**IsocPreqTok: isochronous posted request tokens**. Read-write. Cold reset: 0.
13:12	**IsocReqTok: isochronous request tokens**. Read-write. Cold reset: 0.
11:10	**DnRspTok: downstream response tokens**. Read-write. Cold reset: 1.
9:8	**UpRspTok: upstream response tokens**. Read-write. Cold reset: 2.
7:6	**DnPreqTok: downstream posted request tokens**. Read-write. Cold reset: 1.
5:4	**UpPreqTok: upstream posted request tokens**. Read-write. Cold reset: 1.
3:2	**DnReqTok: downstream request tokens**. Read-write. Cold reset: 1.
1:0	**UpReqTok: upstream request tokens**. 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	**ProbeTok: probe tokens**. Read-write. Cold reset: 3.
3:0	**RspTok: response tokens**. 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.

Bits	Description
31:30	**FreeTok[3:2]: free tokens**. Read-write. See FreeTok[1:0] below.
29	Reserved.
28	**IsocRspTok1: isochronous response tokens sublink 1**. Read-write. Cold reset: 0.
27	Reserved.
26	**IsocPreqTok1: isochronous posted request tokens sublink 1**. Read-write. Cold reset: 0.
25	Reserved.
24	**IsocReqTok1: isochronous request tokens sublink 1**. Read-write. Cold reset: 0.
23:22	**ProbeTok1: probe tokens sublink 1**. Read-write. Cold reset: 0 ganged, 1 unganged.
21:20	**RspTok1: response tokens sublink 1**. Read-write. Cold reset: 0 ganged, 1 unganged.
19:18	**PReqTok1: posted request tokens sublink 1**. Read-write. Cold reset: 0 ganged, 1 unganged.
17:16	**ReqTok1: request tokens sublink 1**. Read-write. Cold reset: 0 ganged, 1 unganged.
15:14	**FreeTok[1:0]: free tokens**. 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	**IsocRspTok0: isochronous response tokens sublink 0**. Read-write. Cold reset: 0.
11:10	**IsocPreqTok0: isochronous posted request tokens sublink 0**. Read-write. Cold reset: 0.
9:8	**IsocReqTok0: isochronous request tokens sublink 0**. Read-write. Cold reset: 0.
7:6	**ProbeTok0: probe tokens sublink 0**. Read-write. Cold reset: 2 ganged, 1 unganged.
5:4	**RspTok0: response tokens sublink 0**. Read-write. Cold reset: 2 ganged, 1 unganged.
3:2	**PReqTok0: posted request tokens sublink 0**. Read-write. Cold reset: 2 ganged, 1 unganged.
1:0	**ReqTok0: request tokens sublink 0**. 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	**LnkToXcsDRToken: display refresh tokens all links**. 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 47 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): Reserved.

For general information on error thresholding, see section 2.13.1.4 [Error Thresholding] on page 119.

Bits	Description
31	**Valid**. Read-only from configuration space; read-only or read-write from MSR space based on MSRC001_0015[McStatusWrEn]. Reset=1b. 1=A valid CntP field is present in this register.
30	**CntP: counter present**. Read-only from configuration space; read-only or read-write from MSR space based on MSRC001_0015[McStatusWrEn]. Reset=1b. 1=A valid threshold counter is present. This bit is affected by MSRC001_0015[McStatusWrEn].
29	**Locked**. Read-only from configuration space; read-only or read-write from MSR space based on MSRC001_0015[McStatusWrEn]. Reset=0b. This bit is set by BIOS to indicate that the 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 MSRC001_0015[McStatusWrEn] is set, MSR writes to this register update all bits, regardless of the state of the Locked bit in the write.
28:24	Reserved.
23:20	**LvtOffset: LVT offset**. Read-write. Reset=0000b. This field 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]). Only values 0 through 3 are valid; all others reserved.
19	**CntEn: counter enable**. Read-write (see also the F3x1[78, 70, 68, 60][Locked]). Reset=0b. 1=Counting of errors specified by Error Threshold Group (see above) is enabled.
18:17	**IntType: interrupt type**. Read-write (see also the F3x1[78, 70, 68, 60][Locked]). Cold reset=X. 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 2.9.2 [CPU Cores and Downcoring] on page 108); see section 2.14.2.3 [SMI Sources And Delivery] on page 128. 11b = Reserved.
16	**Ovrflw: overflow**. Read-write (see also the F3x1[78, 70, 68, 60][Locked]); 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	**ErrCnt: error counter**. Read-write (see also 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
31:23	Reserved.
22	**SyncFloodOnTblWalkErr: sync flood on table walk error enable**. 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	**SyncFloodOnCpuLeakErr: sync flood on CPU leak error enable**. 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	**SyncFloodOnL3LeakErr: sync flood on L3 cache leak error enable**. 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.
10	Reserved.
9	**SyncOnUncNbAryEn: sync flood on uncorrectable NB array error enable**. Read-write. 1=Enables sync flood on detection of an error in a NB array that is uncorrectable.
8	**SyncOnProtEn: sync flood on protocol error enable**. Read-write. 1=Enables sync flood on detection of a protocol error on a link or in the L3.
7	**SyncFloodOnTgtAbtErr**. Read-write. 1=Enable sync flood on generated or received link responses that indicate target aborts.
6	**SyncFloodOnDatErr**. Read-write. 1=Enable sync flood on generated or received link responses that indicate data error.
5	**DisPciCfgCpuMstAbtRsp**. Read-write. 1=Disable MCA error reporting for master abort responses to CPU-initiated configuration accesses. It is recommended that this bit be set in order to avoid MCA exceptions being generated from master aborts for PCI configuration accesses, which are common during device enumeration.
4	**MstAbtChgToNoErrs**. Read-write. 1=Signal no errors instead of master abort in link response packets to IO devices on detection of a master abort condition. When MstAbtChgToNoErrs and F3x44[IoMstAbortDis] are both set, MstAbtChgToNoErrs takes precedence.
3	**DatErrChgToTgtAbt**. Read-write. 1=Signal target abort instead of data error in link response packets to IO devices (for Gen1 link compatibility).
2	**WDTCntSel[3]: watchdog timer count select bit[3]**. Read-write. See F3x44[WDTCntSel].
1	**SyncFloodOnUsPwDataErr: sync flood on upstream posted write data error**. Read-write. 1=Enable sync flood generation when an upstream posted write data error is detected.
0	**McaLogUsPwDataErrEn: MCA log of upstream posted write data error enable**. Read-write. 1=Enable logging of upstream posted write data errors in MCA (if NB MCA registers are appropriately enabled and configured).

F3x190 Downcore Control Register

Cold reset: 0000 0000h. See section 2.9.2 [CPU Cores and Downcoring] on page 108. Changes to this register do not take effect until after a warm reset.

Bits	Description
31:4	Reserved.
3:0	**DisCore[3:0]**. Read-write. 1=Disable the core. 0=Enable the core. Reads provide the last value written regardless of whether there has been a warm reset or not.

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 cores)

If the processor does not include an L3 cache, then L3ToSriReqCBC may be 0h.

Bits	Description
31:15	Reserved.
14:12	**L3ToSriReqCBC: L3 cache to SRI request command buffer count**. Read-write. Cold reset: 4h.
11:9	Reserved.
8:4	**L3FreeListCBC: L3 free list command buffer counter for CPU requests**. Read-write. Cold reset = 1Ch for single-core parts, 18h for dual-core parts, and 10h for quad-core parts. BIOS should set this field up per the following equation (although lower values are legal): • L3FreeListCBC = 32 - CpuCmdBufCnt * (number of cores).
3	Reserved.
2:0	**CpuCmdBufCnt: CPU to SRI command buffer count**. Read-write. Cold reset = 4 if the product includes an L3 cache or 2 if it does not. Each core is allocated the number of buffers specified by this field.

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	**LvtOffsetVal: local vector table offset valid**. 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	**LvtOffset: local vector table offset**. 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 APIC[530:500]). Only values of 03b-00b may be programmed in this field.

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 section 2.13.3 [Sideband Interface (SBI)] on page 123.

Bits	Description
31	**SbiRegWrDn: SBI register write done**. 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.
30:7	Reserved.

6:4	**SbiAddr: SMBus-based sideband interface address**. Read-write. Cold reset: specified by the SA[2:0] strap pins (value matches the pins until the deassertion of RESET_L for a cold reset only; value is not changed by a warm reset); 000b in products that do not include SA[2:0] pins. Specifies bits[3:1] of the SMBus address of the processor SBI ports. SMBus address bits [3:1] = {~SA[2],SA[1:0]}.
3	**SbTsiDis: SMBus-based sideband temperature sensor interface disable**. Read-only. 1=The processor does not support SMBus-based SB-TSI protocol.
2	Reserved.
1	Reserved.
0	Reserved.

F3x1E8 SBI Address Register

Cold reset 0000_0000h. The SB-TSI registers can be directly accessed by the processor using F3x1E8 and F3x1EC. To access the registers, F3x1E8[SbiBankSel and SbiRegAddr] are programmed to point to the appropriate register and the read or write access is directed at F3x1EC.

Bits	Description
31:8	Reserved.
7:0	**SbiRegAddr: SBI SMBus register address**. Read-write. This field specifies the 8-bit address of the SB-TSI register to access.

F3x1EC SBI Data Register

Reset 0000_0000h.

Bits	Description
31:8	Reserved.
7:0	**SbiRegDat: SBI SMBus register data**. Read-write. This field specifies the data to be read or written to the SBI register selected by F3x1E8[SbiRegAddr].

F3x1EC_x[FF:01] SB-TSI Registers

The SB-TSI registers can be accessed by programming F3x1E8[SbiRegAddr] with the offset value. Accesses to reserved offsets result in undefined behavior. 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	Temperature Threshold High Byte Register
09h	SB-TSI Configuration Register
10h	CPU Temperature Low Byte Register
11h	CPU Temperature 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
70h-7Fh	Process Call Registers
80h-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 *SBI Temperature Sensor Interface (SB-TSI) Specification* for register definition.

F3x1F0 Product Information Register

Bits	Description
31:16	Reserved.
15:0	**BrandId**. Read-only. Brand identifier. This is identical to CPUID Fn8000_0001_EBX[BrandId].

F3x1FC Product Information Register

Bits	Description
31:26	Reserved.
25:22	**SinglePlaneNbIdd[3:0].** 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	**DualPlaneNbVidOff[4:0]**. 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 also section 2.4.2.6 [BIOS Northbridge COF and VID Configuration] on page 37.

16:14	**DualPlaneNbFidOff[2:0]**. 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 also section 2.4.2.6 [BIOS Northbridge COF and VID Configuration] on page 37.
13:7	**SinglePlaneNbVid[6:0]**. Read-only. Specifies the NbVid value required for NB operation at the frequency specified by SinglePlaneNbFid. See also section 2.4.2.6 [BIOS Northbridge COF and VID Configuration] on page 37.
6:2	**SinglePlaneNbFid[4:0]**. Read-only. Specifies the NbFid value for platforms with unified VDD and VDDNB power planes. See also section 2.4.2.6 [BIOS Northbridge COF and VID Configuration] on page 37.
1	**NbVidUpdateAll**. Read-only. Indicates that software is required to update NbVid after cold reset based on the sequence defined in section 2.4.2.6 [BIOS Northbridge COF and VID Configuration] on page 37.
0	**NbCofVidUpdate**. Read-only. Indicates that software is required to update the NB COF and NbVid after cold reset based on the sequence defined in section 2.4.2.6 [BIOS Northbridge COF and VID Configuration] on page 37.

3.7 Function 4 Link Control Registers

See section 3.1 [Register Descriptions and Mnemonics] on page 137 for a description of the register naming convention. See section 2.11 [Configuration Space] on page 113 for details about how to access this space.

F4x00 Device/Vendor ID Register

Reset: 1204 1022h.

Bits	Description
31:16	**DeviceID: device ID**. Read-only.
15:0	**VendorID: vendor ID**. Read-only.

F4x04 Status/Command Register

Reset: 00?0 0000h.

Bits	Description
31:16	**Status**. Read-only. Only bit[20] may be set to indicate the existence of a PCI-defined capability block. 0=No supported links are unganged. 1=At least one link may be unganged, in which case there is a capability block associated with sublink one of the link in this function.
15:0	**Command**. Read-only.

F4x08 Class Code/Revision ID Register

Reset: 0600 0000h.

Bits	Description
31:8	**ClassCode**. Read-only. Provides the host bridge class code as defined in the PCI specification.
7:0	**RevID: revision ID**. Read-only.

F4x0C Header Type Register

Reset: 0080 0000h.

Bits	Description
31:0	**HeaderTypeReg**. Read-only. These bits are fixed at their default values. The header type field indicates that there are multiple functions present in this device.

F4x34 Capabilities Pointer Register

Reset: 0000 00??h.

Bits	Description
31:8	Reserved.
7:0	**CapPtr: capabilities pointer**. Read-only. Specifies the offset of the link capabilities block based on which links are supported and unganged. The value provided is: 80h If link 0 is supported and unganged. A0h If link 0 is ganged/unsupported and link 1 is supported and unganged. C0h If link 0 and 1 are ganged/unsupported and link 2 is supported and unganged. E0h If link 0, 1, and 2 are ganged/unsupported and link 3 is supported and unganged.

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.

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.

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.

Bits	Description
31	**LinkPhyDone: link phy access done**. 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	**LinkPhyWrite: link phy read/write select**. 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	**DirectMapEn: direct map enable**. 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	**UpperLinkPhyOffset: upper link phy offset address bits**. Read-write.
9:0	**LinkPhyOffset: link phy offset**. 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	**RttCtl: receiver termination resistance (Rtt) control**. Read-write. Cold reset: 0. This field specifies how the receiver termination resistance value is calculated. All values between 00h and 1Fh are valid. 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. For all modes (except 100b), higher values reduce the resistance of Rtt and lower values increase the resistance of Rtt. See section 2.7.2 [Termination and Compensation] on page 60 for more information about compensation.
28:21	Reserved.
20:16	**RttIndex: receiver termination resistance (Rtt) index**. Read-write. Cold reset: 1_1111b. See RttCtl for details about how this field is used.

15:13	**RonCtl: transmitter resistance (Ron) control**. Read-write. Cold reset: 0. This field specifies how the transmitter resistance value is calculated. 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 aways enabled. This is intended for testing purposes only. 101b - 111b: reserved. For all modes (except 100b), higher values reduce the resistance of Ron and lower values increase the resistance of Ron. See section 2.7.2 [Termination and Compensation] on page 60 for more information about compensation.
12:5	Reserved.
4:0	**RonIndex: transmitter resistance (Ron) index**. Read-write. Cold reset: 1_1111b. See RonCtl for details about how this field is used.

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].

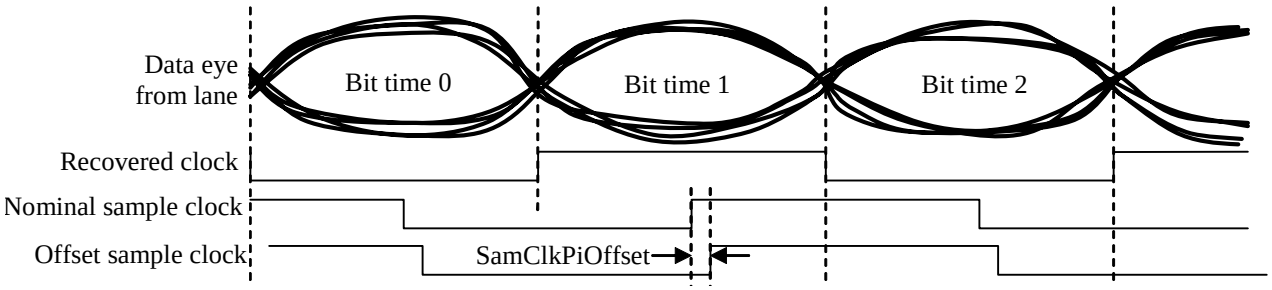

Figure 12: Link phy recovered clock and sample clock.

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	**LfcMax: loop filter counter maximum value**. Read-write. Cold reset: 80h. 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	**LfcMin: loop filter counter minimum value**. Read-write. Cold reset: 40h. 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	**SamClkPiOffsetEn: sample clock phase interpolator offset enable**. Read-write. Cold reset: 0. 1=Enable offset insertion around the nominal sample clock position.
6:4	**SamClkPiOffset: sample clock phase interpolator offset setting**. Read-write. Cold reset: X. This field specifies the magnitude of the offset of the sample clock from the nominal position. See Figure 12. This field is encoded as follows. • **Sample clock phase interpolator offset** = (SamClkPiOffset + 1) * step size. • If link speed is >3.6GT/s, the expected typical step size is 2ps with a +/-1ps error. • If link speed is <=3.6GT/s, the expected typical step size is 3ps with a +/-1ps error.
3	**SamClkPiOffsetSign: sample clock phase interpolator offset setting sign bit**. Read-write. 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 Figure 12.
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	**JitterInjEn: jitter injection enable**. Read-write. Cold reset: 0. 1=Jitter injection test mode is enabled.
29	**JitterInjDir: jitter injection direction**. Read-write. Cold reset: 0. Bit Definition 0 Move clock before the nominal lock position. 1 Move clock after the nominal lock position.
28:23	**JitterInjOnCnt: jitter injection on count**. Read-write. Cold reset: 0.
22:16	Reserved.
15:10	**JitterInjOffCnt:jitter injection off count**. 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	**JitterInjHold:jitter injection hold**. Read-write. Cold reset: 0. 1=Jitter injection hold is enabled.
8:0	Reserved.

F4x1[9C, 94, 8C, 84]_x[D4, C4] Link Phy 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.

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. BIOS should set up the DCV field as follows.

Far-device deemphasis setting	DCV
No deemphasis	20h
-3dB postcursor	17h
-6dB postcursor	10h
-8dB postcursor	0Dh

Bits	Description
31:16	Reserved.
15:10	**DCV: transmit single ended DC voltage level**. 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.
9:8	Reserved.
7:5	Reserved.
4:0	Reserved.

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. See section 2.7.3 [Equalization] on page 60 for more information about deemphasis. These register bits are updated as specified by F0x16C[ImmUpdate].

In Gen3 link DC-coupled mode, three postcursor deemphasis settings, -3dB, -6dB and -8dB are supported. In addition a -11dB postcursor deemphasis setting and a -8dB precursor setting are supported. Normally, the precursor setting is only enabled together with the -11dB postcursor setting. The fields in this register can be programmed during link initialization to select the right deemphasis setting as follows.

Gen3 deemphasis setting	DL1, DL2, DP1	PostCur1En	PostCur2En	PreCur1En	MapPostCur2En
No deemphasis	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
-8dB postcursor	1Fh, 06h, 00h	1	1	0	1
-11dB postcursor	1Fh, 0Dh, 00h	1	1	0	1
-11dB postcursor with -8dB precursor	1Fh, 06h, 07h	1	1	1	1

Note:

- MapPreCurEn=0 for all the supported Gen 3 deemphasis settings.
- Deemphasis is not supported by the transmit clock lanes.

The deemphasis setting for a given platform should be arrived at by modeling the channel response of the platform with different settings using the jEye Platform Development Tool Rev. 1.5.8 or newer. Further adjustment of the settings may be required after validation of the platform due to slight variations between the simulation and the actual platform's channel response.

Deemphasis is not supported when operating at Gen1 link frequencies. Hence, all relevant deemphasis fields in this register should be left in the default state; otherwise, it can cause undefined behavior.

Bits	Description
31	**PostCur1En: post-cursor 1 deemphasis enable**. Read-write. Cold reset: 0. 1=Post-cursor deemphasis is enabled. 0=Post-cursor 1 deemphasis is not supported.
30	**PostCur2En: post-cursor 2 deemphasis enable**. Read-write. Cold reset: 0. 1=Post-cursor deemphasis is enabled. 0=Post-cursor 2 deemphasis is not supported.
29	**PreCur1En: pre-cursor 1 deemphasis enable**. 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	**VML: transmitter voltage margin level**. 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: • Margined diff voltage = nominal diff voltage * (1 - VML/3Eh) 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.
20:16	**DL1: deemphasis level 1**. Read-write. Cold reset: 12h. The cold reset value supports -3dB deemphasis level.
15:13	Reserved.
12:8	**DL2: deemphasis level 2**. Read-write. Cold reset: 0.
7	Reserved.
6	**MapPostCur2En: Map post-cursor 2 deemphasis enable**. Read-write. Cold reset: 0. 1=Post-cursor 2 deemphasis is mapped to post-cursor 1 deemphasis. See above.
5	**MapPreCurEn: Map pre-cursor deemphasis enable**. Read-write. Cold reset: 0. 1=Pre-cursor deemphasis is mapped to post-cursor 1 deemphasis. See above.
4:0	**DP1: deemphasis pre-cursor level 1**. Read-write. Cold reset: 0.

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.
This register should be programmed to 0000_006Dh for Gen 1 links.

Bits	Description
31:8	Reserved.
7:4	**XmtRdPtr: transmit FIFO read pointer**. Read-write. Specified in double-bit time increments. 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.
3:0	**RcvRdPtr: receive FIFO read pointer**. Read-write. Specified in double-bit time increments. 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.

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	**CompCyc: compensation cycle**. Read-write. Cold reset: 0. This specifies the number of internal clock cycles used in averaging out compensation values. Bits Number of clocks 00b 256 01b 128 10b 64 11b 32 It is recommended that these bits remain in the default state.
29:28	Reserved.
27:23	**RttRawCal: receiver termination resistance (Rtt) raw calibration value**. Read-only. Cold reset: X. This field provides the raw Rtt calibration value as determined by the compensation circuit.
22:18	**RonRawCal: transmitter resistance (Ron) raw calibration value**. Read-only. 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	**Width**. 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	**ErrCnt: error count**. 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].
15:13	Reserved.
12:8	**ErrLnNum: error lane number**. 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. 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	**ErrStat: error status**. 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. Bits Status 00b no error 01b training error 10b pattern miscompare 11b reserved
5	**InvRotEn: inversion rotate enable**. 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	**RxDis: receiver disable**. 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	**ConstCnt: constant generator count**. 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	**ConstSel: constant generator select**. Read-write. Cold reset: 0. Selects 0 or 1 to send for the time the constant generator is active.
19:13	**ModCnt: modulo-N count**. 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	**ModSel: modulo-N select**. Read-write. Cold reset: 0. Selects the pattern sent by the Modulo-N counter: 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	**PatCnt: pattern buffer count**. 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	**Order**. Read-write. Cold reset: 0. Selects the order in which each pattern is sent. 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	**Pattern1[23:0]**. 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	**TxMask[8:0]**. Read-write. Cold reset: 1FFh. Selects lanes of the sublinks to transmit a logical 0. 1=Lane active. 0=Lane masked. 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	**TxInv[8:0]**. Read-write. Cold reset: 0. Selects lanes of the sublinks to invert. 1=Lane inverted. 0=Lane unmodified. Bit Lane Bit Lane 0 CAD0 5 CAD5 1 CAD1 6 CAD6 2 CAD2 7 CAD7 3 CAD3 8 CTL 4 CAD4 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.

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	**Pattern2[23:0]**. 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	**Pat2En[8:0]**. 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. 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	**Pattern2[39:24]**. Read-write. Cold reset: 0. Holds the upper 16 bits of Pattern Buffer 2.
15:0	**Pattern1[39:24]**. 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	**TxScramble**. Read-write. Cold reset: 0. Selects lanes of the sublinks to scramble. 1=Scrambling enabled. 0=Scrambling disabled. 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	**ConstCnt: constant generator count**. 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	**ConstSel: constant generator select**. Read-write. Cold reset: 0. Selects 0 or 1 to send for the time the constant generator is active.

19:13	**ModCnt: modulo-N count**. 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	**ModSel: modulo-N select**. Read-write. Cold reset: 0. Selects the pattern sent by the Modulo-N counter: 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	**PatCnt: pattern buffer count**. 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	**Order**. Read-write. Cold reset: 0. Selects the order in which each pattern is sent. 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	**Pattern1[23:0]**. 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	**RxMask[8:0]**. 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. 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	**RxInv[8:0]**. Read-write. Cold reset: 0. Selects lanes of the sublink that are inverted. 1=Lane inverted. 0=Lane unmodified. Bit Lane Bit Lane 0 CAD0 5 CAD5 1 CAD1 6 CAD6 2 CAD2 7 CAD7 3 CAD3 8 CTL 4 CAD4 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.

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	**Pattern2[23:0]**. 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	**Pat2En[8:0]**. 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. 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	**Pattern2[39:24]**. Read-write. Cold reset: 0. Holds the upper 16 bits of Pattern Buffer 2.
15:0	**Pattern1[39:24]**. 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	**RxScramble**. Read-write. Cold reset: 0. Selects lanes of the sublink to scramble. 1=Scrambling enabled. 0=Scrambling disabled. 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
31:9	Reserved.
8:0	**RxErrStat**. 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. 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]_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	**Ls2ExitTime: LS2 exit time**. 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. 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. Note: The value specified by Ls2ExitTime must be less than the value specified by F0x16C[T0Time], or it can cause undefined behavior.
28:18	Reserved.
17	**DllLockFastModeEn: DLL lock fast mode enable**. Read-write. Cold reset: 0. 1=Enables DLL lock fast mode; this is the recommended BIOS setting for mobile platforms only. 0=DLL lock operates at standard speed; this is the recommended setting for all non-mobile platforms.
16:15	Reserved.
14:13	**AnalogWaitTime: analog wait time to turn on DLL**. Read-write. Cold reset: 0. 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. The encodings are as follows: Bits Definition 00b Delay=1.25us. 01b Delay=0.625us. 10b Delay=2.5us. 11b Delay=0.3125us. BIOS should program this field to 10b.
12:11	Reserved.
10	**DllAnalogOkIgnore: DLL analog start signal ignore**. 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.
9:8	Reserved.
7	**BiasDisInLs2: bias disable in LS2 power state**. 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. .
6:5	Reserved.

4	**LockDetOnLs2Exit: DLL lock detect on LS2 exit**. 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	**EnCoreLoopFirst: enable DLL core loop first on LS2 exit**. 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[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	**PstateEn**. Default for [The P-State [4:0] Registers] MSRC001_00[68:64][PstateEn].
26:25	**IddDiv**. Default for [The P-State [4:0] Registers] MSRC001_00[68:64][IddDiv].
24:17	**IddValue**. Default for [The P-State [4:0] Registers] MSRC001_00[68:64][IddValue].
16	**NbDid**. Default for [The P-State [4:0] Registers] MSRC001_00[68:64][NbDid].
15:9	**CpuVid**. Default for [The P-State [4:0] Registers] MSRC001_00[68:64][CpuVid].
8:6	**CpuDid**. Default for [The P-State [4:0] Registers] MSRC001_00[68:64][CpuDid].
5:0	**CpuFid**. Default for [The P-State [4:0] Registers] MSRC001_00[68:64][CpuFid].

3.8 APIC Registers

See section 3.1 [Register Descriptions and Mnemonics] on page 137 for a description of the register naming convention.

APIC20 APIC ID Register

Reset: ??00 0000h.

Bits	Description
31:24	**ApicId**. Read-write. Reset: varies based on core number and node number; see MSRC001_001F[InitApicIdCpuIdLo]. See section 2.9.5.1 [ApicId Enumeration Requirements] on page 110. When F0x68[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 also section 2.9.2 [CPU Cores and Downcoring] on page 108.
23:0	Reserved.

APIC30 APIC Version Register

Reset: 80?? 0010h.

Bits	Description
31	**ExtApicSpace: extended APIC register space present**. Read-only. This bit indicates the presence of extended APIC register space starting at APIC400.

30:24	Reserved.
23:16	**MaxLvtEntry**. Read-only. Reset state varies by product. This field specifies the number of entries in the local vector table minus one.
15:8	Reserved.
7:0	**Version**. Read-only. This field indicates the version number of this APIC implementation.

APIC80 Task Priority Register

Reset: 0000 0000h.

Bits	Description
31:8	Reserved.
7:0	**Priority**. Read-write. This field is assigned by software to set a threshold priority at which the core is interrupted.

APIC90 Arbitration Priority Register

Reset: 0000 0000h.

Bits	Description
31:8	Reserved.
7:0	**Priority**. Read-only. This field indicates the current priority for a pending interrupt, or a task or interrupt being serviced by the core. The priority is used to arbitrate between cores to determine which accepts a lowest-priority interrupt request.

APICA0 Processor Priority Register

Reset: 0000 0000h.

Bits	Description
31:8	Reserved.
7:0	**Priority**. Read-only. This field indicates the core's current priority servicing a task or interrupt, and is used to determine if any pending interrupts should be serviced. It is the higher value of the task priority value and the current highest in-service interrupt.

APICB0 End of Interrupt Register

This register is written by the software interrupt handler to indicate the servicing of the current interrupt is complete.

Bits	Description
31:0	Reserved. Write only. Reads return undefined data.

APICC0 Remote Read Register

Reset: 0000 0000h.

Bits	Description
31:0	**RemoteReadData**. Read-only. This field contains the data resulting from a valid completion of a remote read inter-processor interrupt.

APICD0 Logical Destination Register

Reset: 0000 0000h.

Bits	Description
31:24	**Destination**. Read-write. This field contains this APIC's destination identification. This field is used to determine which interrupts should be accepted.
23:0	Reserved.

APICE0 Destination Format Register

Reset: FFFF FFFFh.

Bits	Description
31:28	**Format**. Read-write. This field controls which format to use when accepting interrupts with a logical destination mode. The allowed values are: • 0h = Cluster destinations are used. • Fh = Flat destinations are used.
27:0	Reserved.

APICF0 Spurious Interrupt Vector Register

Reset: 0000 00FFh.

Bits	Description
31:10	Reserved.
9	**FocusDisable**. Read-write. 1=Disable focus core checking during lowest-priority arbitrated interrupts.
8	**APICSWEn: APIC software enable**. Read-write. 0=SMI, NMI, INIT, Startup and Remote Read interrupts may be accepted; pending interrupts in APIC[170:100] and APIC[270:200] are held, but further fixed, lowest-priority, LINT, and ExtInt interrupts are not accepted. All LVT entry mask bits are set and cannot be cleared.
7:0	**Vector**. Read-write. This field contains the vector that is sent to the core in the event of a spurious interrupt. The behavior of bits 3:0 are controlled as specified by [The Link Transaction Control Register] F0x68[ApicExtSpur].

APIC[170:100] In-Service Registers

Reset: 0000 0000h. The in-service registers provide a bit per interrupt to indicate that the corresponding interrupt is being serviced by the core. APIC100[15:0] are reserved. Interrupts are mapped as follows:

Register	Interrupt Number
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	**InServiceBits**. Read-only. These bits are set when the corresponding interrupt is being serviced by the core.

APIC[1F0:180] Trigger Mode Registers

Reset: 0000 0000h. The trigger mode registers provide a bit per interrupt to indicate the assertion mode of each interrupt. APIC180[15:0] are reserved. Interrupts are mapped as follows:

Register	Interrupt Number
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	**TriggerModeBits**. Read-only. The corresponding trigger mode bit is updated when an interrupt is accepted. The values are: • 0b = edge-triggered interrupt. • 1b = level-triggered interrupt.

APIC[270:200] Interrupt Request Registers

Reset: 0000 0000h. The interrupt request registers provide a bit per interrupt to indicate that the corresponding interrupt has been accepted by the APIC. APIC200[15:0] are reserved. Interrupts are mapped as follows:

Register	Interrupt Number
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	**RequestBits**. 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	**IllegalRegAddr: illegal register address**. Read-write. This bit indicates that an access to a nonexistent register location within this APIC was attempted.
6	**RcvdIllegalVector: received illegal vector**. Read-write. This bit indicates that this APIC has received a message with an illegal vector (00h to 0Fh for fixed and lowest priority interrupts).
5	**SentIllegalVector**. Read-write. This bit indicates that this APIC attempted to send a message with an illegal vector (00h to 0Fh for fixed and lowest priority interrupts).
4	Reserved.
3	**RcvAcceptError: receive accept error**. Read-write. This bit indicates that a message received by this APIC was not accepted by this or any other APIC.
2	**SendAcceptError**. Read-write. This bit indicates that a message sent by this APIC was not accepted by any APIC.
1:0	Reserved.

APIC300 Interrupt Command Register Low

Reset: 0000 0000h. Not all combinations of ICR fields are valid. Only the following combinations are valid:

Table 55: 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	**DestShrthnd: destination shorthand**. Read-write. This field provides a quick way to specify a destination for a message. The valid encodings are as follows: • 00b = Destination field • 01b = Self • 10b = All including self • 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.) If all including self or all excluding self is used, then destination mode is ignored and physical is automatically used.
17:16	**RemoteRdStat: remote read status**. Read-only. The encoding for this field is as follows: • 00b = Read was invalid • 01b = Delivery pending • 10b = Delivery done and access was valid • 11b = Reserved
15	**TM: trigger mode**. Read-write. This bit indicates how this interrupt is triggered. It is defined as follows: • 0 = Edge triggered • 1 = Level triggered
14	**Level**. Read-write. The values for this bit are as follows: • 0 = Deasserted • 1 = Asserted
13	Reserved.
12	**DlvryStat: delivery status**. Read-only. This bit is set to indicate that the interrupt has not yet been accepted by the destination core(s).
11	**DM: destination mode**. Read-write. The values for this bit are as follows: • 0 = Physical • 1 = Logical
10:8	**MsgType**. Read-write. The message types are encoded as follows: • 000b = Fixed • 001b = Lowest Priority • 010b = SMI • 011b = Remote read • 100b = NMI • 101b = INIT • 110b = Startup • 111b = External interrupt
7:0	**Vector**. 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	**DestinationField**. Read-write. This field contains the destination encoding used when APIC300[DestShrthnd] is 00b.
23:0	Reserved.

APIC320 Timer Local Vector Table Entry

Reset: 0001 0000h.

Bits	Description
31:18	Reserved.
17	**Mode**. Read-write. The values for this bit are as follows: • 0 = One-shot • 1 = Periodic
16	**Mask**. Read-write. If this bit is set, this local vector table entry does not generate interrupts.
15:13	Reserved.
12	**DlvryStat: delivery status**. Read-only. This bit is set to indicate that the interrupt has not yet been accepted by the core.
11	Reserved.
10:8	**MsgType: message type**. Write only. Read always returns 000b. See 2.14.1.10 [Generalized Local Vector Table] on page 126 for supported message types.
7:0	**Vector**. Read-write. This field contains the vector that is sent for this interrupt source.

APIC330 Thermal Local Vector Table Entry

Reset: 0001 0000h. Interrupts for this local vector table are caused by transitions in and out of an STC thermal zone as described in 2.10.3.2 [Software Thermal Control (STC)] on page 112, changes in [The P-State Current Limit Register] MSRC001_0061[CurPstateLimit] due to HTC, or STC.

Bits	Description
31:17	Reserved.
16	**Mask**. Read-write. If this bit is set, this local vector table entry does not generate interrupts.
15:13	Reserved.
12	**DlvryStat: delivery status**. Read-only. This bit is set to indicate that the interrupt has not yet been accepted by the core.
11	Reserved.
10:8	**MsgType: message type**. Read-write. See 2.14.1.10 [Generalized Local Vector Table] on page 126 for supported message types.
7:0	**Vector**. Read-write. This field contains the vector that is sent for this interrupt source.

APIC340 Performance Counter Vector Table Entry

Reset: 0001 0000h. Interrupts for this local vector table are caused by overflows of [The Performance Event Counter Registers (PERF_CTR[3:0])] MSRC001_00[07:04]. Note: The Mask bit is not set automatically when the interrupt is taken.

Bits	Description
31:17	Reserved.
16	**Mask**. Read-write. If this bit is set, this local vector table entry does not generate interrupts.
15:13	Reserved.

12	**DlvryStat: delivery status**. Read-only. This bit is set to indicate that the interrupt has not yet been accepted by the core.
11	Reserved.
10:8	**MsgType: message type**. Read-write. See 2.14.1.10 [Generalized Local Vector Table] on page 126 for supported message types.
7:0	**Vector**. Read-write. This field contains the vector that is sent for this interrupt source.

APIC350 Local Interrupt 0 (Legacy INTR) Local Vector Table Entry

Reset: 0001 0000h.

Bits	Description
31:17	Reserved.
16	**Mask**. Read-write. If this bit is set, this local vector table entry does not generate interrupts.
15	**TM: trigger mode**. Read-write. This bit indicates how this interrupt is triggered. It is defined as follows: • 0 = Edge triggered • 1 = Level triggered
14	**RmtIRR**. Read-only. If trigger mode is level, remote IRR is set when the interrupt has begun service. Remote IRR is cleared when the end of interrupt has occurred.
13	**PinPol: pin polarity**. Read-write. This bit is not used because LINT interrupts are delivered by HyperTransport™ messages instead of individual pins.
12	**DlvryStat: delivery status**. Read-only. This bit is set to indicate that the interrupt has not yet been accepted by the core.
11	Reserved.
10:8	**MsgType: message type**. Read-write. See 2.14.1.10 [Generalized Local Vector Table] on page 126 for supported message types.
7:0	**Vector**. Read-write. This field contains the vector that is sent for this interrupt source.

APIC360 Local Interrupt 1(Legacy NMI) Local Vector Table Entry

Reset: 0001 0000h.

Bits	Description
31:0	See APIC350.

APIC370 Error Local Vector Table Entry

Reset: 0001 0000h.

Bits	Description
31:17	Reserved.
16	**Mask**. Read-write. If this bit is set, this local vector table entry does not generate interrupts.
15:13	Reserved.

12	**DlvryStat: delivery status**. Read only. This bit is set to indicate that the interrupt has not yet been accepted by the core.
11	Reserved.
10:8	**MsgType: message type**. Write only. Read always returns 0h. See 2.14.1.10 [Generalized Local Vector Table] on page 126 for supported message types.
7:0	**Vector**. Read-write. This field contains the vector that is sent for this interrupt source.

APIC380 Timer Initial Count Register

Reset: 0000 0000h.

Bits	Description
31:0	**Count**. Read-write. This field contains the value copied into the current count register when the timer is loaded or reloaded.

APIC390 Timer Current Count Register

Reset: 0000 0000h.

Bits	Description
31:0	**Count**. Read only. This field contains the current value of the counter.

APIC3E0 Timer Divide Configuration 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
0	11b	16
1	00b	32
1	01b	64
1	10b	128
1	11b	1

Bits	Description
31:4	Reserved.
3	**Div[3]**. Read-write.
2	Reserved.
1:0	**Div[1:0]**. Read-write.

APIC400 Extended APIC Feature Register

Bits	Description
31:24	Reserved.
23:16	**ExtLvtCount: extended local vector table count**. Read-only, 04h. This specifies the number of extended LVT registers in the local APIC. These registers are [The Extended Interrupt [3:0] Local Vector Table Registers] APIC[530:500].
15:3	Reserved.
2	**ExtApicIdCap: extended APIC ID capable**. Read-only, 1. Indicates that the processor is capable of supporting an 8-bit APIC ID, controlled by APIC410[ExtApicIdEn].
1	**SeioCap: specific end of interrupt capable**. Read-only, 1. This bit indicates that the [The Specific End Of Interrupt Register] APIC420 is present.
0	**IerCap: interrupt enable register capable**. Read-only, 1. This bit indicates that the [The Interrupt Enable Registers] APIC[4F0:480] are present. See 2.14.1.5 [Interrupt Masking] on page 125.

APIC410 Extended APIC Control Register

Reset: 0000 0000h.

Bits	Description
31:3	Reserved.
2	**ExtApicIdEn: extended APIC ID enable**. Read-write. 1=Enable 8-bit APIC ID; APIC20[ApicId] supports an 8-bit value; an interrupt broadcast in physical destination mode requires that the IntDest[7:0]=1111_1111 (instead of xxxx_1111); a match in physical destination mode occurs when (IntDest[7:0] == ApicId[7:0]) instead of (IntDest[3:0] == ApicId[3:0]). Extended APIC ID can also be enabled by writing F0x68[ApicExtId] and F0x68[ApicExtBrdCst].
1	**SeoiEn**. Read-write. This bit enables SEOI generation when a write to the specific end of interrupt register is received.
0	**IerEn**. Read-write. This bit enables writes to the interrupt enable registers.

APIC420 Specific End Of Interrupt Register

Reset: 0000 0000h

Bits	Description
31:8	Reserved.
7:0	**EoiVec: end of interrupt vector**. Read-write. A write to this field causes an end of interrupt cycle to be performed for the vector specified in this field. The behavior is undefined if no interrupt is pending for the specified interrupt vector.

APIC[4F0:480] Interrupt Enable Registers

Reset: FFFF FFFFh

Bits	Description
31:0	**InterruptEnableBits**. Read-write. The interrupt enable bits can be used to enable each of the 256 interrupts. Interrupt enables are mapped as follows: 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: 0000 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	**Mask**. Read-write. 1=This LVT entry does not generate interrupts.
15:13	Reserved.
12	**DlvryStat: delivery status**. Read-only. 1=The interrupt has not yet been accepted by the CPU.
11	Reserved.
10:8	**MsgType: message type**. Read-write. Specifies the interrupt type generated by this LVT entry. See 2.14.1.10 [Generalized Local Vector Table] on page 126 for supported message types.
7:0	**Vector**. 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 section 2.16 [CPUID Instruction] on page 135 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	**LFuncStd**: **largest standard function**. Function 0000_0000h returns the largest CPUID standard-function input value supported by the processor implementation: 0000_0005h. **LFuncExt: largest extended function**. Function 8000_0000h returns the largest CPUID extended-function input value supported by the processor implementation: 8000_001Ah.
EBX, ECX, EDX	31:0	**Vendor: vendor**. 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	**ExtendedFamily**: 01h.
19:16	**ExtendedModel**.
15:12	Reserved.
11:8	**BaseFamily**: Fh.
7:4	**BaseModel**.
3:0	**Stepping**: processor stepping (revision) for a specific model.

CPUID Fn0000_0001_EBX LocalApicId, LogicalProcessorCount, CLFlush, 8BitBrandId

Bits	Description
31:24	**LocalApicId**: initial local APIC physical ID. Provides the initial APIC20[ApicId] value. After F0x60[NodeId] as been initialized, changes to APIC20[ApicId] do not effect the value of this CPUID register. See also section 2.9.2 [CPU Cores and Downcoring] on page 108.
23:16	**LogicalProcessorCount**: 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	**CLFlush**: CLFLUSH size in quadwords = 08h.
7:0	**8BitBrandId**: 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	**PkgType**: package type. Specifies the processor package type. This field is encoded as follows: 0000b: Fr2(1207) or Fr4(1207). 0001b: AM2r2. 0010b: Reserved. 0011b: G3. 01xxb: Reserved. 1xxxb: Reserved
27:16	Reserved.
15:0	**BrandId**: brand ID. This is identical to F3x1F0[BrandId].

CPUID Fn0000_0001_ECX Feature Identifiers

Bits	Description
31:24	Reserved.
23	**POPCNT**: POPCNT instruction = 1.
22:14	Reserved.
13	**CMPXCHG16B**: CMPXCHG16B instruction = 1.
12:4	Reserved.
3	**Monitor**: Monitor/Mwait instructions = 1. This can be disabled through [The Hardware Configuration Register (HWCR)] MSRC001_0015[MonMwaitDis].
2:1	Reserved.
0	**SSE3**: SSE3 extensions = 1; may be overridden by MSRC001_0015[SseDis].

CPUID Fn8000_0001_ECX Feature Identifiers

Bits	Description
31:11	Reserved.
10	**IBS**: Instruction Based Sampling = 1.
9	**OSVW**: OS Visible Work-around support = 1.
8	**3DNowPrefetch**: Prefetch and PrefetchW instructions = 1.
7	**MisAlignSse**: Misaligned SSE Mode = (setting varies by product); may be overridden by MSRC001_0015[MisAlignSseDis].
6	**SSE4A**: EXTRQ, INSERTQ, MOVNTSS, and MOVNTSD instruction support = 1; may be overridden by MSRC001_0015[SseDis].
5	**ABM**: advanced bit manipulation. LZCNT instruction support (setting varies by product).
4	**AltMovCr8**: LOCK MOV CR0 means MOV CR8 = 1.
3	**ExtApicSpace**: extended APIC register space = 1.
2	**SVM**: Secure Virtual Mode feature (setting varies by product).

Bits	Description
1	**CmpLegacy**: core multi-processing legacy mode (setting varies by product). 1=Multi core product (CPUID Fn8000_0008[NC] != 0). 0=Single core product (CPUID Fn8000_0008[NC] = 0).
0	**LahfSahf**: 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	**3DNow**: 3DNow!™ instructions = 1.
30	0000_0001h	Reserved.
	8000_0001h	**3DNowExt**: AMD extensions to 3DNow!™ instructions = 1.
29	0000_0001h	Reserved.
	8000_0001h	**LM**: long mode (may vary by product).
28	0000_0001h	**HTT**: hyper-threading technology (setting varies by product). This bit qualifies the meaning of CPUID Fn0000_0001_EBX[LogicalProcessorCount]. 1=Multi core product (CPUID Fn8000_0008[NC] != 0). 0=Single core product (CPUID Fn8000_0008[NC] = 0).
	8000_0001h	Reserved.
27	0000_0001h	Reserved.
	8000_0001h	**RDTSCP**: RDTSCP instruction = 1.
26	0000_0001h	**SSE2**: SSE2 extensions = 1; may be overridden by MSRC001_0015[SseDis].
	8000_0001h	**Page1GB**: 1 GB large page support = 1.
25	0000_0001h	**SSE**: SSE extensions = 1; may be overridden by MSRC001_0015[SseDis].
	8000_0001h	**FFXSR**: FXSAVE and FXRSTOR instruction optimizations = 1.
24	both	**FXSR**: FXSAVE and FXRSTOR instructions = 1.
23	both	**MMX**: MMX™ instructions = 1.
22	0000_0001h	Reserved.
	8000_0001h	**MmxExt**: AMD extensions to MMX™ instructions = 1.
21	Both	Reserved.
20	0000_0001h	Reserved.
	8000_0001h	**NX**: no-execute page protection = 1.
19	0000_0001h	**CLFSH**: CLFLUSH instruction = 1.
	8000_0001h	Reserved.
18	Both	Reserved.
17	both	**PSE36**: page-size extensions = 1.
16	both	**PAT**: page attribute table = 1.
15	both	**CMOV**: conditional move instructions, CMOV, FCOMI, FCMOV = 1.
14	both	**MCA**: machine check architecture, MCG_CAP = 1.

Bits	Function	Description
13	both	**PGE**: page global extension, CR4.PGE = 1.
12	both	**MTRR**: memory-type range registers = 1.
11	0000_0001h	**SysEnterSysExit**: SYSENTER and SYSEXIT instructions = 1.
	8000_0001h	**SysCallSysRet**: SYSCALL and SYSRET instructions = 1.
10	both	Reserved.
9	both	**APIC**: advanced programmable interrupt controller (APIC) exists and is enabled. This bit reflects the state of [The APIC Base Address Register (APIC_BAR)] MSR0000_001B[ApicEn].
8	both	**CMPXCHG8B**: CMPXCHG8B instruction = 1.
7	both	**MCE**: machine check exception, CR4.MCE = 1.
6	both	**PAE**: physical-address extensions (PAE) = 1.
5	both	**MSR**: AMD model-specific registers (MSRs), with RDMSR and WRMSR instructions = 1.
4	both	**TSC**: time stamp counter, RDTSC/RDTSCP instructions, CR4.TSD = 1.
3	both	**PSE**: page-size extensions (4 MB pages) = 1.
2	both	**DE**: debugging extensions, IO breakpoints, CR4.DE = 1.
1	both	**VME**: virtual-mode enhancements = 1.
0	both	**FPU**: 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	**IBE**: Interrupt break-event = 1.
ECX	0	**EMX:** 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.
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 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	**L2DTlb2and4MAssoc**. L2 data TLB associativity for 2 MB and 4 MB pages = 2.
EAX	27:16	**L2DTlb2and4MSize**. 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	**L2ITlb2and4MAssoc**. L2 instruction TLB associativity for 2 MB and 4 MB pages = 0.
EAX	11:0	**L2ITlb2and4MSize**. L2 instruction TLB number of entries for 2 MB and 4 MB pages = 0.
EBX	31:28	**L2DTlb4KAssoc**. L2 data TLB associativity for 4 KB pages = 4.
EBX	27:16	**L2DTlb4KSize**. L2 data TLB number of entries for 4 KB pages = 512.
EBX	15:12	**L2ITlb4KAssoc**. L2 instruction TLB associativity for 4 KB pages = 4.
EBX	11:0	**L2ITlb4KSize**. L2 instruction TLB number of entries for 4 KB pages = 512.
ECX	31:16	**L2Size**. L2 cache size in KB (varies with product). May be one of 256, 512, or 1024.
ECX	15:12	**L2Assoc**. L2 cache associativity = 8.
ECX	11:8	**L2LinesPerTag**. L2 cache lines per tag = 1.
ECX	7:0	**L2LineSize**. L2 cache line size in bytes = 64.
EDX	31:18	**L3Size**. L3 cache size (varies with product). L3 cache size is at least (L3Size[31:18] * 512KB) and less than ((L3Size[31:18] +1) * 512KB).
EDX	17:16	Reserved.
EDX	15:12	**L3Assoc**. L3 cache associativity = (varies with product); supported values are 16, 32, 48, and 64.
EDX	11:8	**L3LinesPerTag**. L3 cache lines per tag = 1.
EDX	7:0	**L3LineSize**. 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	31:9	Reserved.
EDX	8	**TscInvariant**: TSC rate is invariant = 1.

Register	Bits	Description
EDX	7	**HwPstate**: 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	**100MHzSteps**: 100 MHz multiplier Control = 1.
EDX	5	**STC**: software thermal control (STC) is supported (support may vary by product).
EDX	4	**TM**: hardware thermal control (HTC) is supported (support may vary by product).
EDX	3	**TTP**: THERMTRIP is supported = 1.
EDX	2	**VID**: Voltage ID control is supported = 0 (function replaced by HwPstate).
EDX	1	**FID**: Frequency ID control is supported = 0 (function replaced by HwPstate).
EDX	0	**TS**: 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 CPUID Fn[8000_0001, 0000_0001]_EDX[LM]) 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	**ApicIdCoreIdSize[3:0]**. The number of bits in the initial APIC20[ApicId] value that indicate core ID within a processor = 2h.
ECX	11:8	Reserved.
ECX	7:0	**NC: number of physical cores - 1**. 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 F3x190[DisCore]. See also section 2.9.2 [CPU Cores and Downcoring] on page 108.
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	**SvmRev**: SVM revision = 01h.
EBX	31:0	**NASID**: number of address space identifiers (ASID) = 40h.

Register	Bits	Description
ECX	31:0	Reserved.
EDX	31:4	Reserved.
EDX	3	**NRIPS**: NRIP Save= 0.
EDX	2	**SVML**: SVM lock = 1.
EDX	1	**LbrVirt**: LBR virtualization = 1.
EDX	0	**NP**: 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 = 0.
EBX	27:16	L2 data TLB number of entries for 1 GB pages = 0.
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	**MOVU**.
EAX	0	**FP128**.
EBX	31:0	Reserved.
ECX	31:0	Reserved.
EDX	31:0	Reserved.

3.10 MSRs - MSR0000_xxxx

See section 3.1 [Register Descriptions and Mnemonics] on page 137 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	**TSC: time stamp counter**. 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.

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	**ApicBar: APIC base address register**. Read-write. Specifies the base address for the APICXX register set. See section 3.8 [APIC Registers] on page 275 for details about this register set.
11	**ApicEn: APIC enable**. 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	**BSC: boot strap core**. Read-write. 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	**ClusterID**. Read-only. This is normally 00b; the value does not affect hardware.
15:0	Reserved.

MSR0000_00FE MTRR Capabilities Register (MTRRcap)

Reset: 0000 0000 0000 0508h.

Bits	Description
63:11	Reserved.

10	**MtrrCapWc: write-combining memory type**. Read-only. 1=The write combining memory type is supported.
9	Reserved.
8	**MtrrCapFix: fixed range register**. Read-only. 1=Fixed MTRRs are supported.
7:0	**MtrrCapVCnt: variable range registers count**. Read-only. Specifies the number of variable MTRRs supported.

MSR0000_0174 SYSENTER CS Register (SYSENTER_CS)

Reset: 0000 0000 0000 0000h.

Bits	Description
63:32	RAZ.
31:16	SBZ.
15:0	**SYSENTER_CS: SYSENTER target CS**. Read-write. Holds the called procedure code segment.

MSR0000_0175 SYSENTER ESP Register (SYSENTER_ESP)

Reset: 0000 0000 0000 0000h.

Bits	Description
63:32	Reserved.
31:0	**SYSENTER_ESP: SYSENTER target SP**. Read-write. Holds the called procedure stack pointer.

MSR0000_0176 SYSENTER EIP Register (SYSENTER_EIP)

Reset: 0000 0000 0000 0000h.

Bits	Description
63:32	Reserved.
31:0	**SYSENTER_EIP: SYSENTER target IP**. Read-write. Holds the called procedure instruction pointer.

MSR0000_0179 Global Machine Check Capabilities Register (MCG_CAP)

Reset: 0000 0000 0000 0106h.

Bits	Description
63:9	Reserved
8	**MCG_CTL_P: MCG_CTL register present**. Read-only. 1=The machine check control registers (MCi_CTL; see section 2.13.1 [Machine Check Architecture] on page 114) are present.
7:0	**Count**. Read-only. Indicates the number of error-reporting banks visible to each core.

MSR0000_017A Global Machine Check Status Register (MCG_STAT)

Reset: 0000 0000 0000 0000h. See also 2.13.1 [Machine Check Architecture] on page 114.

Bits	Description
63:3	Reserved.

2	**MCIP: machine check in progress**. Read-write; set-by-hardware. 1=A machine check is in progress.
1	**EIPV: error instruction pointer valid**. Read-write; updated-by-hardware. 1=The instruction pointer that was pushed onto the stack by the machine check mechanism references the instruction that caused the machine check error.
0	**RIPV: restart instruction pointer valid**. Read-write; updated-by-hardware. 1=Program execution can be reliably restarted at the EIP address on the stack.

MSR0000_017B Global Machine Check Exception Reporting Control Register (MCG_CTL)

Reset: 0000 0000 0000 0000h. See also 2.13.1 [Machine Check Architecture] on page 114. It is expected that this register is programmed to the same value in all nodes.

Bits	Description
63:6	Reserved
5	**FRE: fixed issue reorder buffer register bank enable**. Read-write. 1=The fixed-issue reorder buffer machine check register bank is enabled.
4	**NBE: Northbridge register bank enable**. Read-write. 1=The Northbridge machine check register bank is enabled.
3	**LSE: load-store register bank enable**. Read-write. 1=The load/store machine check register bank is enabled.
2	**BUE: bus unit register bank enable**. Read-write. 1=The bus unit machine check register bank is enabled.
1	**ICE: instruction cache register bank enable**. Read-write. 1=The instruction cache machine check register bank is enabled.
0	**DCE: data cache register bank enable**. 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	**PB: performance monitor pin control**. Read-write. This field does not control any hardware.
1	**BTF**. Read-write. 1=Enable branch single step.
0	**LBR**. Read-write. 1=Enable last branch record.

MSR0000_01DB Last Branch From IP Register (BR_FROM)

Bits	Description
63:0	**LastBranchFromIP**. Read-only. Loaded with the segment offset of the branch instruction.

MSR0000_01DC Last Branch To IP Register (BR_TO)

Bits	Description
63:0	**LastBranchToIP**. 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	**LastIntFromIP**. 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	**LastIntToIP**. Read-only. Holds the target RIP of the last branch that occurred before the exception or interrupt.

MSR0000_02[0F:00] Variable-Size MTRRs (MTRRphysBase*n* and MTRRphysMask*n*)

Reset: xxxx xxxx xxxx xxxxh. 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 also section 2.9.3 [Access Type Determination] on page 108. 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 also section 2.9.3 [Access Type Determination] on page 108. 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.
  1h = WC or write combining.
  4h = WT or write through.
  5h = WP or write protect.
  6h = WB or write back.
  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:

CPUAddr[47:12] & PhyMask[47:12] == PhyBase[47:12] & 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] (MTRRphysBase*n*)

Bits	Description
63:48	MBZ.
47:12	**PhyBase: base address**. Read-write.
11:8	MBZ.
7:0	**MemType: memory type**. Read-write.

MSR0000_020[F, D, B, 9, 7, 5, 3, 1] (MTRRphysMask*n*)

Bits	Description
63:48	MBZ.
47:12	**PhyMask: address mask**. Read-write.
11	**Valid**. Read-write. 1=The variable-size MTRR pair is enabled.
10:0	MBZ.

MSR0000_02[6F:68, 59, 58, 50] Fixed-Size MTRRs (MTRRfix*n*)

Reset: xxxx xxxx xxxx xxxxh. 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	**MemType: memory type**. Read-write. Address range from 7_0000 to 7_FFFF.
55:48	**MemType: memory type**. Read-write. Address range from 6_0000 to 6_FFFF.
47:40	**MemType: memory type**. Read-write. Address range from 5_0000 to 5_FFFF.
39:32	**MemType: memory type**. Read-write. Address range from 4_0000 to 4_FFFF.
31:24	**MemType: memory type**. Read-write. Address range from 3_0000 to 3_FFFF.
23:16	**MemType: memory type**. Read-write. Address range from 2_0000 to 2_FFFF.
15:8	**MemType: memory type**. Read-write. Address range from 1_0000 to 1_FFFF.
7:0	**MemType: memory type**. 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	**MemType: memory type**. Read-write. Address range from 1_C000 to 1_FFFF (plus the base).
55:48	**MemType: memory type**. Read-write. Address range from 1_8000 to 1_BFFF (plus the base).
47:40	**MemType: memory type**. Read-write. Address range from 1_4000 to 1_7FFF (plus the base).
39:32	**MemType: memory type**. Read-write. Address range from 1_0000 to 1_3FFF (plus the base).
31:24	**MemType: memory type**. Read-write. Address range from 0_C000 to 0_FFFF (plus the base).
23:16	**MemType: memory type**. Read-write. Address range from 0_8000 to 0_BFFF (plus the base).
15:8	**MemType: memory type**. Read-write. Address range from 0_4000 to 0_7FFF (plus the base).
7:0	**MemType: memory type**. 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	**MemType: memory type**. Read-write. Address range from 0_7000 to 0_7FFF (plus the base).
55:48	**MemType: memory type**. Read-write. Address range from 0_6000 to 0_6FFF (plus the base).
47:40	**MemType: memory type**. Read-write. Address range from 0_5000 to 0_5FFF (plus the base).
39:32	**MemType: memory type**. Read-write. Address range from 0_4000 to 0_4FFF (plus the base).
31:24	**MemType: memory type**. Read-write. Address range from 0_3000 to 0_3FFF (plus the base).
23:16	**MemType: memory type**. Read-write. Address range from 0_2000 to 0_2FFF (plus the base).
15:8	**MemType: memory type**. Read-write. Address range from 0_1000 to 0_1FFF (plus the base).
7:0	**MemType: memory type**. 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.
1h = WC or write combining.
4h = WT or write through.
All other values result in a #GP(0).
5h = WP or write protect.
6h = WB or write back.
7h = UC- or uncacheable (overridden by MTRR WC state)

Bits	Description
63:59	MBZ.
58:56	**PA7 MemType**. Read-write. Default UC. MemType for {PAT, PCD, PWT} = 7h.

55:51	MBZ.
50:48	**PA6 MemType**. Read-write. Default UC-. MemType for {PAT, PCD, PWT} = 6h.
47:43	MBZ.
42:40	**PA5 MemType**. Read-write. Default WT. MemType for {PAT, PCD, PWT} = 5h.
39:35	MBZ.
34:32	**PA4 MemType**. Read-write. Default WB. MemType for {PAT, PCD, PWT} = 4h.
31:27	MBZ.
26:24	**PA3 MemType**. Read-write. Default UC. MemType for {PAT, PCD, PWT} = 3h.
23:19	MBZ.
18:16	**PA2 MemType**. Read-write. Default UC-. MemType for {PAT, PCD, PWT} = 2h.
15:11	MBZ.
10:8	**PA1 MemType**. Read-write. Default WT. MemType for {PAT, PCD, PWT} = 1h.
7:3	MBZ.
2:0	**PA0 MemType**. 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	**MtrrDefTypeEn: variable and fixed MTRR enable**. Read-write. 1=[The Variable-Size MTRRs (MTRRphysBasen and MTRRphysMaskn)] MSR0000_02[0F:00], and [The Fixed-Size MTRRs (MTRRfixn)] MSR0000_02[6F:68, 59, 58, 50], are enabled. 0=Fixed and variable MTRRs are not enabled.
10	**MtrrDefTypeFixEn: fixed MTRR enable**. Read-write. 1=[The Fixed-Size MTRRs (MTRRfixn)] MSR0000_02[6F:68, 59, 58, 50] are enabled. This field is ignored (and the fixed MTRRs are not enabled) if MSR0000_02FF[MtrrDefTypeEn]=0.
9:8	MBZ.
7:0	**MemType: memory type**. Read-write. Specifies the memory type for space not mapped to enabled [The Variable-Size MTRRs (MTRRphysBasen and MTRRphysMaskn)] MSR0000_02[0F:00], or enabled [The Fixed-Size MTRRs (MTRRfixn)] MSR0000_02[6F:68, 59, 58, 50].

MSR0000_0400 DC Machine Check Control Register (MC0_CTL)

Reset: 0000 0000 0000 0000h. All defined bits are read-write.

See section 2.13.1 [Machine Check Architecture] on page 114. For all bits, 1=Enable the specified reporting mechanism.

Bits	Enable
63:7	Reserved.
6	**L2TP: L2 TLB parity errors**. Report data cache L2 TLB parity errors.
5	**L1TP: L1 TLB parity errors**. Report data cache L1 TLB parity errors.

4	**DSTP: snoop tag array parity errors**. Report data cache snoop tag array parity errors.
3	**DMTP: main tag array parity errors**. Report data cache main tag array parity errors.
2	**DECC: data array ECC errors**. Report data cache data array ECC errors. If not set, ECC errors in the cache are detected and logged, but not reported. If masked (see MSRC001_00[49:44]), ECC errors in the cache are undetected.
1	**ECCM: multi-bit ECC data errors**. Report multi-bit ECC data errors during data cache line fills from the internal L2 or the system. If masked (see MSRC001_00[49:44]), multi-bit ECC errors on line fills may be detected and logged as single-bit errors unless single-bit ECC data errors are also masked (ECCI). If masking all line fill data errors is desired, all ECC data error mask bits (ECCI and ECCM) must be set.
0	**ECCI: single-bit ECC data errors**. Report single-bit ECC data errors during data cache line fills from the internal L2 or the system. If masked (see MSRC001_00[49:44]), multi-bit ECC errors on line fills may also be masked. If masking all line fill data errors is desired, all ECC data error mask bits (ECCI and ECCM) must be set.

MSR0000_0401 DC Machine Check Status Register (MC0_STATUS)

Cold reset: xxxx xxxx xxxx xxxxh. See section 2.13.1 [Machine Check Architecture] on page 114. 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 also MSRC001_0015[McStatusWrEn]. The following field definitions apply to all MCi_STATUS registers, except as noted.

Bits	Description
63	**Val: valid**. Read-write; set-by-hardware. 1=A valid error has been detected (whether it is enabled or not). This bit should be cleared to 0 by software after the register has been read.
62	**Over: error overflow**. Read-write; set-by-hardware. 1=An error was detected while the valid bit (Val) of this register was set; at least one error was not logged. The machine check mechanism handles the contents of MCi_STATUS during overflow as outlined in section 2.13.1.2.2 [Machine Check Error Logging Overwrite During Overflow] on page 117.
61	**UC: error uncorrected**. Read-write; updated-by-hardware. 1=The error was not corrected by hardware.
60	**En: error enable**. Read-write; updated-by-hardware. 1=MCA error reporting is enabled for this error in MCi_CTL.
59	**MiscV: miscellaneous error register valid**. Read-only. 1=MCi_MISC contains valid information for this error. This bit is always 0, except in the case of [The NB Machine Check Misc (Thresholding) Register (MC4_MISC0)] MSR0000_0413 and [The FR Machine Check Miscellaneous Register (MC5_MISC)] MSR0000_0417.
58	**AddrV: error address valid**. Read-write; updated-by-hardware. 1=The address saved in MCi_ADDR is the address where the error occurred.
57	**PCC: processor context corrupt**. Read-write; updated-by-hardware. 1=The state of the processor may have been corrupted by the error condition. Restart may not be reliable.
56:55	Reserved.
54:47	**Syndrome[7:0]**. Read-write. • MC0_STATUS (DC): The lower eight syndrome bits when an ECC error is detected. See Table 33 for the mappings that show which bit errors result in which syndrome values. • MC[3:1]_STATUS (LS, BU, IC): Reserved.

46	**CECC: correctable ECC error**. Read-write; updated-by-hardware. 1=The error was a correctable ECC error.
45	**UECC: uncorrectable ECC error**. Read-write; updated-by-hardware. 1=The error was an uncorrectable ECC error.
44:41	Reserved.
40	**Scrub: error detected on a scrub**. Read-write; updated-by-hardware. • MC0_STATUS (DC), MC2_STATUS (BU): 1=The error was detected on a scrub. • MC1_STATUS (IC), MC3_STATUS (LS), MC5_STATUS (FR): Reserved.
39:32	Reserved.
31:24	**Syndrome[15:8]**. Read-write. • MC0_STATUS (DC): The upper eight syndrome bits when an ECC error is detected. See Table 33 for the mappings that show which bit errors result in which syndrome values. • MC[3:1]_STATUS (LS, BU, IC): Reserved.
23:20	Reserved.
19:16	**ErrorCodeExt: extended error code**. Read-write. See the appropriate error signature tables below: • MC0_STATUS (DC): Table 57 • MC1_STATUS (IC): Table 60 • MC2_STATUS (BU): Table 63 • MC3_STATUS (LS): Table 65 • MC5_STATUS (FR): Table 66
15:0	**ErrorCode: error code**. Read-write. See the appropriate error signature tables below: • MC0_STATUS (DC): Table 57 • MC1_STATUS (IC): Table 60 • MC2_STATUS (BU): Table 63 • MC3_STATUS (LS): Table 65 • MC5_STATUS (FR): Table 66

This register reports these DC errors:

Table 56: DC error descriptions

Error Type	Description	Enablers (MSR0000_0400 Control Bits)
L2 Cache Line Fill	An error occurred during an L1 line fill from the L2 cache.	ECC1, ECCM.
Data Load/ Store/ Victim/ Snoop	A data error occurred while accessing or managing data.	DECC
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

Table 56: DC error descriptions

Error Type	Description	Enablers (MSR0000_0400 Control Bits)
L1 TLB Multi-match	Hit multiple entries.	L1TP
L2 TLB	Parity error in L2 TLB.	L2TP
L2 TLB Multi-match	Hit multiple entries.	L2TP

Table 57: 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	[44] Reserved	[43] Reserved	[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 multi-bit	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 xxxxh. See section 2.13.1 [Machine Check Architecture] on page 114. 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 58 defines the address register as a function of error type.

Table 58: DC error data; address register

Error Type	Memory Transaction Type (RRRR; Table 43)	Address Register Bits	Description
L2 Cache Line Fill	DRD	47:6	Physical address
Data Load/ Store/ Victim/ Snoop	DRD	47:4[1]	Physical address
	DWR		
	Evict	11:6	Physical address
	Snoop		
Data Scrub	GEN	11:4	Physical address
Tag Snoop/ Victim	Snoop	11:6	Physical address
	Evict		
Tag Load/ Store	DRD	11:6[2]	Physical address
	DWR		
L1 TLB	-	47:12	Linear address
L1 TLB Multi-match			
L2 TLB			
L2 TLB Multi-match			

1. For Data Store (DWR), address bits shown are present only if error was reported (MSR0000_0401[UC] is set and MSR0000_0400[DECC] is enabled and not masked). If not reported, then valid address register bits are the linear address in 14:4.
2. The entire address from the TLB may be stored, but that address may only be incidentally related to the tag error; only the indicated bits are valid for this type of error.

MSR0000_0403 DC Machine Check Miscellaneous Register (MC0_MISC)

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 section 2.13.1 [Machine Check Architecture] on page 114. For all bits, 1=Enable the specified reporting mechanism.

Bits	Enable
63:10	Reserved.
9	**RDDE: read data errors**. Report system read data errors for an instruction cache fetch if [The BU Machine Check Control Register (MC2_CTL)] MSR0000_0408[SRDE_ALL] = 1.
8:7	Reserved.
6	**L2TP: L2 TLB parity errors**. Report instruction cache L2 TLB parity errors.

Bits	Enable
5	**L1TP: L1 TLB parity errors**. Report instruction cache L1 TLB parity errors.
4	**ISTP: snoop tag array parity errors**. Report instruction cache snoop tag array parity errors.
3	**IMTP: main tag array parity errors**. Report instruction cache main tag array parity errors.
2	**IDP: data array parity errors**. Report instruction cache data array parity errors.
1	**ECCM: multi-bit ECC data errors**. Report multi-bit ECC data errors during instruction cache line fills or TLB reloads from the internal L2 or the system.
0	**ECCI: single-bit ECC data errors**. Report single-bit ECC data errors during instruction cache line fills or TLB reloads from the internal L2 or the system.

MSR0000_0405 IC Machine Check Status Register (MC1_STATUS)

Cold reset: xxxx xxxx xxxx xxxxh. See section 2.13.1 [Machine Check Architecture] on page 114. See also MSR0000_0401 for the information about all of the MCi_STATUS registers. See also MSRC001_0015[McStatusWrEn]. This register reports these IC errors:

Table 59: 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 60: 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	[44] Reserved	[43] Reserved	[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	Mem-ory	-	-	IRD	Instr	L2	0[1]	1	0	N	0[2]	1			0
IC Data Load (Parity)	0000	Mem-ory	-	-	IRD	Instr	L1	0	1	0	N	0	0			0
Tag Snoop	0000	Mem-ory	-	-	Snoop	Instr	L1	1	1	1	N	0	0			0
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

1. Line refetched from memory. (Automatically purged from L2 during fill.)
2. Single bit errors are detected as parity errors.

MSR0000_0406 IC Machine Check Address Register (MC1_ADDR)

Cold reset: xxxx xxxx xxxx xxxxh. See section 2.13.1 [Machine Check Architecture] on page 114. 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 61 defines the address register as a function of error type.

Table 61: 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 47:20 for 2-Mbyte page	Linear address
L1 TLB Multi-match		
L2 TLB	47:12 for 4-Kbyte page	Linear address
L2 TLB Multi-match		

MSR0000_0407 IC Machine Check Miscellaneous Register (MC1_MISC)

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 section 2.13.1 [Machine Check Architecture] on page 114. For all bits, 1=Enable the specified reporting mechanism.

Bits	Enable
63:12	Reserved.
11	**PDC_PAR: Pdc/GTLB parity errors**. Report Page Descriptor Cache parity or Guest TLB table walk parity errors.
10	**VB_PAR: write/victim data buffer parity error**. Report write buffer or victim buffer data parity errors.
9	Reserved.
8	**L2D_UECC: L2 data uncorrectable ECC error**. Report L2 data array uncorrectable ECC errors.
7	**L2D_CECC: L2 data correctable ECC error**. Report L2 data array correctable ECC errors.
6	**L2D_PAR: L2 data parity errors**. Report correctable and uncorrectable L2 data array parity errors.
5	**L2T_UECC: L2 tag uncorrectable ECC error**. Report L2 tag array uncorrectable ECC errors.
4	**L2T_CECC: L2 tag correctable ECC error**. Report L2 tag array correctable ECC errors.
3	**L2T_PAR: L2 tag parity errors**. Report L2 tag array correctable and uncorrectable parity errors.
2	**SRDE_ALL: all system read data**. Report system read data errors for any operation including a DC/IC fetch, TLB reload or hardware prefetch.
1	**SRDE_TLB: system read data TLB reload**. Report system read data errors for a TLB reload.
0	**SRDE_HP: system read data hardware prefetch**. Report system read data errors for a hardware prefetch.

MSR0000_0409 BU Machine Check Status Register (MC2_STATUS)

Cold reset: xxxx xxxx xxxx xxxxh. See section 2.13.1 [Machine Check Architecture] on page 114. See also MSR0000_0401 for the information about all of the MCi_STATUS registers. See also MSRC001_0015[McStatusWrEn]. This register reports these BU errors:

Table 62: 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

Table 62: BU error descriptions

Error Type	Description	Enablers (MSR0000_0408 Control Bits)
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 63: 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	[44] Reserved	[43] Reserved	[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 xxxxh. See section 2.13.1 [Machine Check Architecture] on page 114. 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 64 defines the address register as a function of error type

Table 64: 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 14:6 for 512-Kbyte L2 13:6 for 256-Kbyte L2 12:6 for 128-Kbyte L2	Physical address
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 section 2.13.1 [Machine Check Architecture] on page 114. For all bits, 1=Enable the specified reporting mechanism.

Bits	Enable
63:2	Reserved.
1	**SRDE_S: read data errors on store**. Report system read data errors on a store if [The BU Machine Check Control Register (MC2_CTL)] MSR0000_0408[SRDE_ALL] = 1.
0	**SRDE_L: read data errors on load**. Report system read data errors on a load if [The BU Machine Check Control Register (MC2_CTL)] MSR0000_0408[SRDE_ALL] = 1.

MSR0000_040D LS Machine Check Status Register (MC3_STATUS)

Cold reset: xxxx xxxx xxxx xxxxh. See section 2.13.1 [Machine Check Architecture] on page 114. See also MSR0000_0401 for the information about all of the MCi_STATUS registers. See also MSRC001_0015[McStatusWrEn]. This register reports these LS errors:

Table 65: 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	[44] Reserved	[43] Reserved	[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 xxxxh. See section 2.13.1 [Machine Check Architecture] on page 114. 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 section 2.13.1 [Machine Check Architecture] on page 114. 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 section 3.1.1 [Northbridge MSRs In Multi-Core Products] on page 138.

Bits	Enable
63:32	Reserved.
31:0	See [The MCA NB Control Register] F3x40.

MSR0000_0411 NB Machine Check Status Register (MC4_STATUS)

See section 2.13.1 [Machine Check Architecture] on page 114. 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 section 3.1.1 [Northbridge MSRs In Multi-Core Products] on page 138.

Bits	Enable
63:32	See F3x4C.
31:0	See F3x48.

MSR0000_0412 NB Machine Check Address Register (MC4_ADDR)

See section 2.13.1 [Machine Check Architecture] on page 114. 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 section 3.1.1 [Northbridge MSRs In Multi-Core Products] on page 138.

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 (Thresh-

olding) Registers 1 to 3 (MC4_MISC[3:1])] MSRC000_04[0A:08]. For general information on error thresholding, see section 2.13.1.4 [Error Thresholding] on page 119.

Only one of these registers exists in multi-core devices; see section 3.1.1 [Northbridge MSRs In Multi-Core Products] on page 138.

Bits	Enable
63:32	See F3x1[78, 70, 68, 60].
31:24	**BlkPtr: Block pointer for additional MISC registers**. Read-only. Valid only when Valid field set. When non-zero, used to calculate a pointer to the extended MISC MSR block (see Section 2.13.1.1 [Machine Check Registers] on page 115) 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 section 2.13.1 [Machine Check Architecture] on page 114. For all bits, 1=Enable the specified reporting mechanism.

Bits	Enable
63:1	Reserved.
0	**CPUWDT: CPU watchdog timer**. The CPU 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 xxxxh. See section 2.13.1 [Machine Check Architecture] on page 114. See also MSR0000_0401 for the information about all of the MCi_STATUS registers. See also MSRC001_0015[McStatusWrEn]. This register reports these FR errors:

Table 66: 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	[44] Reserved	[43] Reserved	[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 xxxxh. See section 2.13.1 [Machine Check Architecture] on page 114. 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)

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	**FFXSE: fast FXSAVE/FRSTOR enable**. Read-write. 1=Enables the fast FXSAVE/FRSTOR mechanism. A 64-bit operating system uses CPUID Fn[8000_0001, 0000_0001]_EDX[24] to determine the presence of this feature before enabling it. This bit is set once by the operating system and its value is not changed afterwards.
13	**LMSLE: long mode segment limit enable**. Read-write. 1=Enables the long mode segment limit check mechanism.
12	**SVME: secure virtual machine (SVM) enable**. Read-write. 1=SVM features are enabled.
11	**NXE: no-execute page enable**. Read-write. 1=The no-execute page protection feature is enabled.
10	**LMA: long mode active**. Read-only. 1=Indicates that long mode is active.
9	MBZ.
8	**LME: long mode enable**. Read-write. 1=Long mode is enabled.
7:1	RAZ.
0	**SYSCALL: system call extension enable**. Read-write. 1=SYSCALL and SYSRET instructions are enabled. This adds the SYSCALL and SYSRET instructions which can be used in flat addressed operating systems as low latency system calls and returns.

MSRC000_0081 SYSCALL Target Address Register (STAR)

Reset: X. This register holds the target address used by the SYSCALL instruction and the code and stack segment selector bases used by the SYSCALL and SYSRET instructions.

Bits	Description
63:48	**SysRetSel: SYSRET CS and SS**. Read-write.
47:32	**SysCallSel: SYSCALL CS and SS**. Read-write.
31:0	**Target: SYSCALL target address**. Read-write.

MSRC000_0082 Long Mode SYSCALL Target Address Register (STAR64)

Reset: X.

Bits	Description
63:0	**LSTAR: long mode target address**. Read-write. Target address for 64-bit mode calling programs. The address stored in this register must be in canonical form (if not canonical, a #GP fault occurs).

MSRC000_0083 Compatibility Mode SYSCALL Target Address Register (STARCOMPAT)

Reset: X.

Bits	Description
63:0	**CSTAR: compatibility mode target address**. Read-write. Target address for compatibility mode. The address stored in this register must be in canonical form (if not canonical, a #GP fault occurs).

MSRC000_0084 SYSCALL Flag Mask Register (SYSCALL_FLAG_MASK)

Reset: X.

Bits	Description
63:32	RAZ.
31:0	**MASK: SYSCALL flag mask**. Read-write. This register holds the EFLAGS mask used by the SYSCALL instruction. 1=Clear the corresponding EFLAGS bit when executing the SYSCALL instruction.

MSRC000_0100 FS Base Register (FS_BASE)

Reset: X.

Bits	Description
63:0	**FS_BASE: expanded FS segment base**. Read-write. This register provides access to the expanded 64-bit FS segment base. The address stored in this register must be in canonical form (if not canonical, a #GP fault fill occurs).

MSRC000_0101 GS Base Register (GS_BASE)

Reset: X.

Bits	Description
63:0	**GS_BASE: expanded GS segment base**. Read-write. This register provides access to the expanded 64-bit GS segment base. The address stored in this register must be in canonical form (if not canonical, a #GP fault fill occurs).

MSRC000_0102 Kernel GS Base Register (KernelGSbase)

Reset: X.

Bits	Description
63:0	**KernelGSBase: kernel data structure pointer**. Read-write. This register holds the kernel data structure pointer which can be swapped with the GS_BASE register using the SwapGS instruction. The address stored in this register must be in canonical form (if not canonical, a #GP fault occurs).

MSRC000_0103 Auxiliary Time Stamp Counter Register (TSC_AUX)

Reset: 0000 0000 0000 0000h.

Bits	Description
63:32	Reserved.
31:0	**TscAux: auxiliary time stamp counter data**. Read-write. It is expected that this is initialized by privileged software to a meaningful value, such as a processor ID. This value is returned in the RDTSCP instruction.

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 section 3.1.1 [Northbridge MSRs In Multi-Core Products] on page 138.

Bits	Description
63:32	See F3x1[78, 70, 68, 60][31:0].
31:0	See MSR0000_0413[31:0].

3.12 MSRs - MSRC001_0xxx

MSRC001_00[03:00] Performance Event Select Register (PERF_CTL[3:0])

Reset: xxxx xxxx xxxx xxxxh. 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] on page 339 shows the events and unit masks supported by the processor.

To accurately start counting with the write that enables the counter, disable the counter when changing the event and then enable the counter with a second MSR write.

The edge count mode increments the counter when a transition happens on the monitored event. If the event selected is changed without disabling the counter, an extra edge is falsely detected when the first event is a static 0 and the second event is a static one. To avoid this false edge detection, disable the counter when changing the event and then enable the counter with a second MSR write.

The performance counter registers can be used to track events in the Northbridge. Northbridge events include all memory controller events, 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	**HostOnly: host only counter**. Read-write. 1=Events are only counted when the processor is in host mode.
40	**GuestOnly: guest only counter**. Read-write. 1=Events are only counted when the processor is in guest mode.
39:36	Reserved
35:32	**EventSelect[11:8]: performance event select**. Read-write. See EventSelect[7:0].
31:24	**CntMask: counter mask**. Read-write. Controls the number of events counted per clock cycle. 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	**Inv: invert counter mask**. Read-write. See CntMask.
22	**En: enable performance counter**. Read-write. 1= Performance event counter is enabled.
21	Reserved
20	**Int: enable APIC interrupt**. Read-write. 1=APIC performance counter LVT interrupt is enabled to generate an interrupt when the performance counter overflows.
19	Reserved.
18	**Edge: edge detect**. Read-write. 0=Level detect. 1=Edge detect.
17	**OS: OS mode**. Read-write. 1=Events are only counted when CPL=0.
16	**User: user mode**. Read-write. 1=Events only counted when CPL>0.

15:8	**UnitMask: event qualification**. Read-write. Each UnitMask bit further specifies or qualifies the event specified by EventSelect. All events selected by UnitMask are simultaneously monitored. Unless otherwise stated, the UnitMask values shown may be combined (logically ORed) to select any desired combination of the sub-events for a given event. In some cases, certain combinations can result in misleading counts, or the UnitMask value is an ordinal rather than a bit mask. These situations are described where applicable, or should be obvious from the event descriptions. For events where no UnitMask table is shown, the UnitMask is not applicable and may be set to zeros.
7:0	**EventSelect[7:0]: event select**. Read-write. This field, along with EventSelect[11:8] above, combine to form the 12-bit event select field, EventSelect[11:0]. EventSelect specifies the event or event duration in a processor unit to be counted by the corresponding PERF_CTR[3:0] register. The events are specified in section 3.14 [Performance Counter Events] on page 339. Some events are reserved; when a reserved event is selected, the results are undefined.

MSRC001_00[07:04] Performance Event Counter Registers (PERF_CTR[3:0])

Reset: 0000 xxxx xxxx xxxxh. 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	**CTR: performance counter value**. Read-only. Returns the current value of the event counter.

MSRC001_0010 System Configuration Register (SYS_CFG)

Reset: 0000 0000 0002 0600h.

Bits	Description
63:23	Reserved.
22	**Tom2ForceMemTypeWB: top of memory 2 memory type write back**. Read-write. 1=The default memory type of memory between 4GB and TOM2 is write back instead of the memory type defined by [The MTRR Default Memory Type Register (MTRRdefType)] MSR0000_02FF[MemType]. For this bit to have any effect, MSR0000_02FF[MtrrDefTypeEn] must be 1. MTRRs and PAT can be used to override this memory type.
21	**MtrrTom2En: MTRR top of memory 2 enable**. Read-write. 0=[The Top Of Memory 2 Register (TOM2)] MSRC001_001D is disabled. 1=This register is enabled.
20	**MtrrVarDramEn: MTRR variable DRAM enable**. Read-write. 0=[The Top Of Memory Register (TOP_MEM)] MSRC001_001A and IORRs are disabled. 1=These registers are enabled. This bit should be set by BIOS.
19	**MtrrFixDramModEn: MTRR fixed RdDram and WrDram modification enable**. Read-write. 0=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	**MtrrFixDramEn: MTRR fixed RdDram and WrDram attributes enable**. 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	**SysUcLockEn: system lock command enable**. Read-write. 1=Transactions to the coherent fabric support the lock command. This is normally enabled in multi-core systems and disabled in single core systems.
16	**ChxToDirtyDis: change to dirty disable**. Read-write. 1=Disables change-to-dirty commands, evicts line from DC instead.
15:11	Reserved.
10	**SetDirtyEnO: clean-to-dirty command for O->M state transition enable**. Read-write. 1=Enables generating write probes when transitioning a cache line from Owned to Modified.
9	**SetDirtyEnS: shared-to-dirty command for S->M state transition enable**. Read-write. 1=Enables generating write probes when transitioning a cache line from Shared to Modified.
8	**SetDirtyEnE: shared-to-dirty command for E->M state transition enable**. Read-write. 1=Enables generating write probes when transitioning a cache line from Exclusive to Modified.
7:0	Reserved.

MSRC001_0015 Hardware Configuration Register (HWCR)

Reset: 0000 0000 0000 0010h.

Bits	Description
63:25	Reserved.

24	**TscFreqSel: TSC frequency select**. Read-write. 0=The TSC increments at the rate of the NCLK frequency. 1=The TSC increments at the rate of the core P-state 0 COF specified by MSRC001_0064 at the time this bit is set by software. Changing the state of this bit after setting it results in undefined behaviour from the TSC. Changing the state of MSRC001_0064 after setting this bit has no effect on the TSC rate. BIOS should program this bit to 1.
23	**ForceUsRdWrSzPrb: force probes for upstream RdSized and WrSized**. 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	**MisAlignSseDis: misaligned SSE mode disable**. Read-write. 1=Disables misaligned SSE mode. If this is set, then CPUID Fn8000_0001_ECX[MisAlignSse] is 0.
20	**IoCfgGpFault: IO-space configuration causes a GP fault**. Read-write. 1=IO-space accesses to configuration space cause a GP fault. The fault is triggered if any part of the IO read/write address range is between CF8h and CFFh, inclusive. These faults only result from single IO instructions, not to string and REP IO instructions. This fault takes priority over the IO trap mechanism described by [The IO Trap Control Register (SMI_ON_IO_TRAP_CTL_STS)] MSRC001_0054.
19	Reserved.
18	**McStatusWrEn: machine check status write enable**. Read-write. 1=Writes by software to MCi_STATUS (see section 2.13.1 [Machine Check Architecture] on page 114) 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]. McStatusWrEn can be used to debug machine check interrupt handlers. See section 2.13.1.6 [Error Injection and Simulation] on page 120.
17	**Wrap32Dis: 32-bit address wrap disable**. Read-write. 1=Disable 32-bit address wrapping. Software can use Wrap32Dis to access physical memory above 4 Gbytes without switching into 64-bit mode. To do so, software should write a greater-than 4 Gbyte address to [The FS Base Register (FS_BASE)] MSRC000_0100 and [The GS Base Register (GS_BASE)] MSRC000_0101. Then it would address ±2 Gbytes from one of those bases using normal memory reference instructions with a FS or GS override prefix. However, the INVLPG, FST, and SSE store instructions generate 32-bit addresses in legacy mode, regardless of the state of Wrap32Dis.
16	Reserved.
15	**SseDis: SSE instructions disable**. Read-write. 1=Disables SSE instructions. If this is set, then CPUID Fn[8000_0001, 0000_0001]_EDX[SSE, SSE2], CPUID Fn0000_0001_ECX[SSE3], and CPUID Fn8000_0001_ECX[SSE4A] are 0.
14	**RsmSpCycDis: RSM special bus cycle disable**. Read-write; read-only if SmmLock=1. 0=A link special bus cycle, SMIACK, is generated on a resume from SMI.
13	**SmiSpCycDis: SMI special bus cycle disable**. Read-write; read-only if SmmLock=1. 0=A link special bus cycle, SMIACK, is generated when an SMI interrupt is taken.
12	**HltXSpCycEn: halt-exit special bus cycle enable**. Read-write. 1=A link special bus cycle is generated when exiting from the halt state.
11	**LimitCpuidStdMaxVal**. Read-write. 1=Limit CPUID standard maximum value, returned by CPUID Fn0000_0000_EAX (see CPUID Fn[8000_0000, 0000_0000]), to 1.

10	**MonMwaitUserEn: MONITOR/MWAIT user mode enable**. Read-write. 1=The MONITOR and MWAIT instructions are supported in all privilege levels. 0=The MONITOR and MWAIT instructions are supported only in privilege level 0; these instructions in privilege levels 1 to 3 cause a #UD exception. The state of this bit is ignored if MonMwaitDis is set.
9	**MonMwaitDis: MONITOR and MWAIT disable**. Read-write. 1=The MONITOR and MWAIT opcodes become invalid. This affects what is reported back through CPUID Fn0000_0001_ECX[Monitor].
8	**IgnneEm: IGNNE port emulation enable**. Read-write. 1=Enable emulation of IGNNE port.
7:5	Reserved.
4	**INVD_WBINVD: INVD to WBINVD conversion**. Read-write. 1=Convert INVD to WBINVD. This bit is required to be set when the L3 cache is enabled.
3	**TlbCacheDis: cacheable memory disable**. Read-write. 1=Disable performance improvement that assumes that the PML4, PDP, PDE and PTE entries are in cacheable memory. Operating systems that maintain page tables in uncacheable memory (UC memory type) must set the TlbCacheDis bit to insure proper operation.
2	SBZ.
1	**SlowFence: slow SFENCE enable**. Read-write. 1=Enable slow sfence.
0	**SmmLock: SMM code lock**. 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	**PhyBase: physical base address**. Read-write.
11:5	RAZ.
4	**RdMem: read from memory**. Read-write. 1=Read accesses to the range are directed to system memory. 0=Read accesses to the range are directed to IO.
3	**WrMem: write to memory**. Read-write. 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	**PhyMask: physical address mask**. Read-write.
11	**Valid**. 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	**TOM[47:23]: top of memory**. 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 section 2.9.3 [Access Type Determination] on page 108.
22:0	RAZ.

MSRC001_001D Top Of Memory 2 Register (TOM2)

Reset: X.

Bits	Description
63:48	RAZ.
47:23	**TOM2[47:23]: second top of memory**. 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 section 2.9.3 [Access Type Determination] on page 108. 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 section 3.1.1 [Northbridge MSRs In Multi-Core Products] on page 138.

Bits	Description
63:55	Reserved.
54	**InitApicIdCpuIdLo**. Read-write. 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]}. See section 2.9.2 [CPU Cores and Downcoring] on page 108 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	**DisOrderRdRsp**. Read-write. 1=Disables ordered responses to IO link read requests. See section 2.7.8 [Response Ordering] on page 62.

49:47	Reserved.
46	**EnableCf8ExtCfg: enable CF8 extended configuration cycles**. Read-write. 1=Allows the IO configuration space access method, IOCF8 and IOCFC, to be used to generate extended configuration cycles by enabling IOCF8[27:24].
45	**DisUsSysMgtReqToNcHt: disable upstream system management request to link**. Read-write. 1=Disables downstream reflection of upstream STPCLK and x86 legacy input system management commands (in order to work around potential deadlock scenarios related to reflection regions).
44:37	Reserved
36	**DisDatMsk: disable data mask**. 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	**DisCohLdtCfg: disable coherent link configuration accesses**. 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	**DisXdsBypass: disable xbar data scheduler bypass**. 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	**DisRefUseFreeBuf: disable display refresh to use free list buffers**. 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	**RedirSmiEn**. Read-write. 1=Redirect MCEs (that are directed to this core) to generate an SMI-trigger IO cycle via MSRC001_0056. The status is stored in SMMFEC4[MceRedirSts].
8	**RedirVecEn**. Read-write. 1=Redirect MCEs (that are directed to this core) to generate a vectored interrupt, using the interrupt vector specified in RedirVector.
7:0	**RedirVector**. Read-write. See RedirVecEn.

MSRC001_00[35:30] Processor Name String Registers

Reset: 0000 0000 0000 0000h. These registers holds the CPUID name string in ASCII. The state of these registers are returned by CPUID instructions, CPUID Fn8000_000[4, 3, 2]. BIOS should set these registers to the product name for the processor as provided by AMD. Each register contains a block of 8 ASCII characters; the least byte corresponds to the first ASCII character of the block; the most-significant byte corresponds to the last character of the block. MSRC001_0030 contains the first block of the name string; MSRC001_0035 contains the last block of the name string.

Bits	Description
63:0	**CpuNameString**. Read-write.

MSRC001_00[49:44] Machine Check Control Mask Registers (MC*i*_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 section 3.1.1 [Northbridge MSRs In Multi-Core Products] on page 138. BIOS is recommended to mask HT retries (F3x40[RtryHt3En, RtryHt2En, RtryHt1En, RtryHt0En]) if the OS is not capable of distinguishing that HT retries are normal operation.
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	**MSK: Control Register Masks**. 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 section 2.13.1 [Machine Check Architecture] on page 114. 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	**SmiOnRdEn: enable SMI on IO read**. Read-write. 1=Enables SMI generation on a read access.
62	**SmiOnWrEn: enable SMI on IO write**. Read-write. 1=Enables SMI generation on a write access.
61	**ConfigSmi: configuration space SMI**. Read-write. 1=Configuration access. 0=IO access (that is not an IO-space configuration access).
60:56	SBZ.
55:32	**SmiMask[23:0]**. Read-write. SMI IO trap mask. 0=Mask address bit. 1=Do not mask address bit.
31:0	**SmiAddr[31:0]**. Read-write. SMI IO trap address.

MSRC001_0054 IO Trap Control Register (SMI_ON_IO_TRAP_CTL_STS)

Reset: 0000 0000 0000 0000h. For each of the SmiEn bits below, 1=The trap specified by the corresponding MSR is enabled. See also MSRC001_00[53:50].

Bits	Description
63:32	RAZ.
31:16	SBZ.
15	**IoTrapEn: IO trap enable**. Read-write. 1=Enable IO and configuration space trapping specified by MSRC001_00[53:50] and MSRC001_0054.
14:8	SBZ.
7	**SmiEn_3: SMI enable for the trap specified by MSRC001_0053**. Read-write.
6	SBZ.
5	**SmiEn_2: SMI enable for the trap specified by MSRC001_0052**. Read-write.
4	SBZ.
3	**SmiEn_1: SMI enable for the trap specified by MSRC001_0051**. Read-write.
2	SBZ.
1	**SmiEn_0: SMI enable for the trap specified by MSRC001_0050**. Read-write.
0	SBZ.

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 pending 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.
31:28	SBZ.
27	**SmiOnCmpHalt: SMI on chip multi-processing halt**. 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 IntPndMsg must be 0, otherwise the behavior is undefined. The status is stored in SMMFEC4[SmiOnCmpHaltSts].
26	**IORd: IO Read**. Read-write. 1=IO read; 0=IO write.
25	**IntrPndMsg: interrupt pending message**. Read-write. 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].
24	**IntrPndMsgDis: interrupt pending message disable**. Read-write. Disable generating the interrupt pending message specified by IntrPndMsg.
23:16	**IOMsgData: IO message data**. Read-write. IO write message data. This field is only used if IORd specifies an IO write message.
15:0	**IOMsgAddr: IO message address**. Read-write. IO space message address.

MSRC001_0056 SMI Trigger IO Cycle Register

Reset: 0000 0000 0000 0000h. See section 2.14.2.3 [SMI Sources And Delivery] on page 128. This register specifies an IO cycle that may be generated when a local SMI trigger event occurs. If IoCycleEn is set and there is a local SMI trigger event, then the IO cycle generated is a byte read or write, based on IoRd, to address IoPortAddress. If the cycle is a write, then IoData contains the data written. If the cycle is a read, the value read is discarded. If IoCycleEn is clear and a local SMI trigger event occurs, then undefined behavior results.

Bits	Description
63:27	Reserved.
26	**IoRd: IO Read**. Read-write. 1=IO read; 0=IO write.
25	**IoCycleEn: IO cycle enable**. Read-write. 1=The SMI trigger IO cycle is enabled to be generated.
24	Reserved.
23:16	**IoData**. Read-write.
15:0	**IoPortAddress**. Read-write.

MSRC001_0058 MMIO Configuration Base Address Register

Reset: xxxx xxxx xxxx xxx0h. See section 2.11 [Configuration Space] on page 113 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	**MmioCfgBaseAddr[47:20]: MMIO configuration base address bits[47:20]**. 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	**BusRange:bus range identifier**. 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: 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	**Enable**. 1=MMIO configuration space is enabled.

MSRC001_0061 P-State Current Limit Register

See also section 2.4.2 [P-states] on page 34. Writes to this register cause a #GP.

Bits	Description
63:7	RAZ.
6:4	**PstateMaxVal: P-state maximum value**. Read-only. Specifies the lowest performance P-state (highest value) supported by the hardware. The state of this field is controlled through [The Clock Power/Timing Control 2 Register] F3xDC[PstateMaxVal].
3	RAZ.
2:0	**CurPstateLimit: current P-state limit**. Read-only. Provides the lowest-performance P-state limit (highest value) from [The Hardware Thermal Control (HTC) Register] F3x64[HtcPstateLimit], and [The Software Thermal Control (STC) Register] F3x68[StcPstateLimit].

MSRC001_0062 P-State Control Register

Bits	Description
63:3	MBZ.
2:0	**PstateCmd: P-state change command**. 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 P-state number, specified by MSRC001_00[68:64]. 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 section 2.4.2 [P-states] on page 34. Reads from this field return the last written value, regardless of whether any limits are applied.

MSRC001_0063 P-State Status Register

Writes to this register cause a #GP.

Bits	Description
63:3	RAZ.
2:0	**CurPstate: current P-state**. Read-only. Cold reset: values vary by product. This field provides the frequency component of the current P-state of the core (regardless of the source of the P-state change, including MSRC001_0062[PstateCmd]; see section 2.4.2.4 [P-state Transition Behavior] on page 35 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. See section 2.4.2 [P-states] on page 34.

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. See section 2.4.2 [P-states] on page 34 for more information about these registers.

Bits	Description
63	**PstateEn**. Read-write. 1=The P-state specified by this MSR is valid. 0=The P-state specified by this MSR is not valid. The purpose of this register is to indicate if the rest of the P-state information in the register is valid after a reset; it controls no hardware.
62:42	SBZ.
41:40	**IddDiv: current divisor field**. Read-write. See MSRC001_00[68:64][IddValue].
39:32	**IddValue: current value field**. Read-write. After a reset, IddDiv and IddValue combine to specify the expected current dissipation of a single core that is in the P-state corresponding to the MSR number. These values are intended to be used to create ACPI-defined _PSS objects (see section 2.4.2.9 [ACPI Processor P-State Objects] on page 47) and to perform the 2.4.2.7 [Processor-System-board Power Delivery Compatibility Check] on page 39. 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: IddDiv — Current Equation — Current Range 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	**NbVid: Northbridge VID**. Read-write. See section 2.4.1 [Processor Power Planes And Voltage Control] on page 28. 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 the NB and cores.
24:23	SBZ.
22	**NbDid: Northbridge divisor ID**. Read-write. Specifies the NB frequency divisor; see F3xD4[NbFid]. 0=Divisor of 1. 1=Divisor of 2. This bit must be set the same in all P-state registers.
21:16	SBZ.
15:9	**CpuVid: CPU core VID**. Read-write. See section 2.4.1 [Processor Power Planes And Voltage Control] on page 28. This field is required to be programmed as specified by MSRC001_0071[MaxVid and MinVid] (otherwise undefined behavior results).
8:6	**CpuDid: CPU core divisor ID**. Read-write. Specifies the CPU frequency divisor; see CpuFid. 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	**CpuFid: CPU core frequency ID**. Read-write. Specifies the CPU frequency multiplier. The CPU COF specified by CpuFid and CpuDid is: CPU COF = 100 MHz * (CpuFid + 10h) / (2^CpuDid). This field and CpuDid must be programmed to the requirements specified in MSRC001_0071[MaxCpuCof]. The value of this field must be less than or equal to 2Fh.

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 also section 2.4.2 [P-states] on page 34.

Bits	Description
63:32	RAZ.
31:25	**NbVid: Northbridge VID**. Read-write. See MSRC001_00[68:64].
24:23	RAZ.
22	**NbDid: Northbridge divisor ID**. Read-only. See MSRC001_00[68:64].
21:19	RAZ.
18:16	**PstateId: P-state identifier**. 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.
15:9	**CpuVid: CPU core VID**. Read-write. See MSRC001_00[68:64].
8:6	**CpuDid: CPU core divisor ID**. Read-write. See MSRC001_00[68:64]. The PstateId field must be updated to cause a new CpuDid value to take effect.
5:0	**CpuFid: CPU core frequency ID**. Read-write. See MSRC001_00[68:64]. The PstateId field must be updated to cause a new CpuFid value to take effect.

MSRC001_0071 COFVID Status Register

See section 2.4.2 [P-states] on page 34.

Bits	Description
63:59	**MaxNbFid: maximum NB COF**. 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	**CurPstateLimit: current P-state limit**. Read-only. Provides the current highest-performance P-state limit (lowest value). Identical to MSRC001_0061[CurPstateLimit].
55	Reserved.
54:49	**MaxCpuCof: maximum CPU COF**. 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	**MinVid: minimum voltage**. 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 section 2.4.1 [Processor Power Planes And Voltage Control] on page 28.
41:35	**MaxVid: maximum voltage**. 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 section 2.4.1 [Processor Power Planes And Voltage Control] on page 28.
34:32	**StartupPstate: startup P-state number**. Read-only. Specifies the cold reset VID, FID and DID for the NB and core based on the P-state number selected (see MSRC001_00[68:64]). If F3xA0[PviMode]=1, then NbVid of the selected P-state is applied to the PVI. Note: if F3xA0[CofVidProg]=0, then the state of this field is ignored, the VID, FID, and DID are applied to the core and NB as specified by that bit.
31:25	**CurNbVid: current Northbridge VID**. Read-only.
24:23	Reserved.
22	**CurNbDid: current Northbridge divisor ID**. Read-only.
21:19	Reserved.
18:16	**CurPstate: current P-state**. Read-only. This is identical to MSRC001_0063[CurPstate].
15:9	**CurCpuVid: current CPU core VID**. Read-only.
8:6	**CurCpuDid: current CPU core divisor ID**. Read-only.
5:0	**CurCpuFid: current CPU core frequency ID**. Read-only.

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	**CpuWdtCountSel: CPU watchdog timer count select**. Read-write. This, along with CpuWdtTimeBase, specifies the time period required for the WDT to expire. The time period is the value specified here times the time base specified by CpuWdtTimeBase. 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	**CpuWdtTimeBase: CPU watchdog timer time base**. Read-write. Specifies the time base for the timeout period specified in CpuWdtCountSel. 00b = 1 millisecond. 10b = 5 nanoseconds. 01b = 1 microsecond. 11b = Reserved.
0	**CpuWdtEn: CPU watchdog timer enable**. Read-write. 1=The WDT is enabled.

MSRC001_0111 SMM Base Address Register (SMM_BASE)

Reset: 0000 0000 0003 0000h. This holds the base of the SMM memory region. The value of this register is stored in the save state on entry into SMM (see section 2.14.2.5 [SMM Save State] on page 129) 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	**SMM_BASE**. Read-write; read-only if MSRC001_0015[SmmLock]=1.

MSRC001_0112 SMM TSeg Base Address Register (SMMAddr)

Reset: 0000 0000 0000 0000h. See section 2.14.2 [System Management Mode (SMM)] on page 127 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:

CPUAddr[47:17] & TSegMask[47:17] == TSegBase[47:17] & 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	**TSegBase[47:17]: TSeg address range base**. Read-write; read-only if MSRC001_0015[SmmLock]=1.
16:0	Reserved.

MSRC001_0113 SMM TSeg Mask Register (SMMMask)

Reset: 0000 0000 0000 0000h. See section 2.14.2 [System Management Mode (SMM)] on page 127 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	**TSegMask[47:17]: TSeg address range mask**. Read-write; read-only if MSRC001_0015[SmmLock]=1. See MSRC001_0112.
16:15	Reserved.
14:12	**TMTypeDram: TSeg address range memory type**. Read-write; read-only if MSRC001_0015[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 MSR0000_02[0F:00].
11	Reserved.
10:8	**AMTypeDram: ASeg Range Memory Type**. Read-write; read-only if MSRC001_0015[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 MSR0000_02[0F:00].
7:6	Reserved.

5	**TMTypeIoWc: non-SMM TSeg address range memory type**. Read-write; read-only if MSRC001_0015[SmmLock]=1. Specifies the attribute of TSeg accesses that are directed to MMIO space. 0=UC (uncacheable). 1=WC (write combining).
4	**AMTypeIoWc: non-SMM ASeg address range memory type**. Read-write; read-only if MSRC001_0015[SmmLock]=1. Specifies the attribute of ASeg accesses that are directed to MMIO space. 0=UC (uncacheable). 1=WC (write combining).
3	**TClose: send TSeg address range data accesses to MMIO**. Read-write. 1=When in SMM, direct data accesses in the TSeg address range to MMIO space. See also, AClose.
2	**AClose: send ASeg address range data accesses to MMIO**. Read-write. 1=When in SMM, direct data accesses in the ASeg address range to MMIO space. [A, T]Close allows the SMI handler to access the MMIO space located in the same address region as the [A, T]Seg. When the SMI handler is finished accessing the MMIO space, it must clear the bit. Failure to do so before resuming from SMM causes the CPU to erroneously read the save state from MMIO space.
1	**TValid: enable TSeg SMM address range**. Read-write; read-only if MSRC001_0015[SmmLock]=1. 1=The TSeg address range SMM enabled.
0	**AValid: enable ASeg SMM address range**. Read-write; read-only if MSRC001_0015[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	**Svme_Disable: SVME disable**. See Lock. 1=MSRC000_0080[SVME] must be zero (MBZ) when writing to MSRC000_0080. Setting this bit when MSRC000_0080[SVME]=1 causes a #GP fault, regardless of the state of Lock. 0=MSRC000_0080[SVME] is read-write.
3	**Lock: SVM lock**. Read-only; write-1-only (see MSRC001_0118). 1=Svme_Disable is read-only. 0=Svme_Disable is read-write.
2	**dis_a20m: disable A20 masking**. Read-write; set-by-hardware. 1=Disables A20 masking. This bit is set by hardware when the SKINIT instruction is executed.
1	**r_init: intercept INIT**. Read-write; set-by-hardware. This bit controls how INIT is delivered in host mode. This bit is set by hardware when the SKINIT instruction is executed. 0 = INIT delivered normally. 1 = INIT translated into a SX interrupt.
0	**dpd: debug port disable**. Read-write; set-by-hardware. This bit controls if debug facilities such as JTAG 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	**IGNNE: current IGNNE state**. Read-write. Reset: X. This bit controls the current state of the processor internal IGNNE signal.

MSRC001_0116 SMM Control Register (SMM_CTL)

Accesses to this register cause a #GP if MSRC001_0015[SmmLock]=1. 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	Reserved.
4	**rsm_cycle: send RSM special cycle**. Write-only. Reset: X. 1=Send a RSM special cycle.
3	**smm_exit: exit SMM**. Write-only. Reset: X. 1=Exit SMM.
2	**smi_cycle: send SMI special cycle**. Write-only. Reset: X. 1=Send a SMI special cycle.
1	**smm_enter: enter SMM**. Write-only. Reset: X. 1=Enter SMM.
0	**smm_dismiss: clear SMI**. 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	**VM_HSAVE_PA: physical address of host save area**. Read-write. This register contains the physical address where VMRUN saves host state and where vm-exit restores host state from. Writing this register causes a #GP if any of the lower 12 bits are not zero or if the address written is greater than FD_000_000h.

MSRC001_0118 SVM Lock Key

Reset: 0000 0000 0000 0000h.

Bits	Description
63:0	**SvmLockKey: SVM lock key**. RAZ, write-only. Writes to this register when MSRC001_0114[Lock]=0 write the SvmLockKey. Writes to this register when MSRC001_0114[Lock]=1 and SvmLockKey!=0 cause hardware to clear MSRC001_0114[Lock] if the value written is the same as the value stored in SvmLockKey.

MSRC001_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 [The Local SMI Status] SMMFEC4.

MSRC001_0140 OS Visible Work-around MSR0 (OSVW_ID_Length)

Reset: 0000 0000 0000 0000h.

Bits	Description
63:16	Reserved.
15:0	**OSVW_ID_Length: OS visible work-around ID length**. Read-write. See the *Revision Guide for AMD Family 10h Processors* for the definition of this field.

MSRC001_0141 OS Visible Work-around MSR1 (OSVW Status)

Reset: 0000 0000 0000 0000h.

Bits	Description
63:0	**OsvwStatusBits: OS visible work-around status bits**. Read-write. See the *Revision Guide for AMD Family 10h Processors* 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	**Features**. Read-write. Provides back-door control over the features reported in CPUID function 1, ECX (see CPUID Fn0000_0001_ECX).
31:0	**Features**. Read-write. Provides back-door control over the features reported in CPUID function 1, EDX (see CPUID Fn[8000_0001, 0000_0001]_EDX).

MSRC001_1005 Extended CPUID Features Register (ExtFeatures)

See also MSRC001_1004.

Bits	Description
63:32	**ExtFeaturesEcx**. Read-write. Provides back-door control over the features reported in CPUID function 8000_0001, ECX (see CPUID Fn8000_0001_ECX).
31:0	**ExtFeaturesEdx**. Read-write. Provides back-door control over the features reported in CPUID function 8000_0001, EDX (see CPUID Fn[8000_0001, 0000_0001]_EDX).

MSRC001_1022 Data Cache Configuration Register (DC_CFG)

All defined fields are read-write.

Bits	Description
63:36	Reserved.
35:34	**REQ_CTR**. Reset: 11b. Initial number of requests (1 to 3) a hardware prefetch can make. Setting this field to 0 disables the hardware prefetcher. The BIOS should program this to 01b for server products.
33:0	Reserved.

MSRC001_1023 Bus Unit Configuration Register (BU_CFG)

Revision B2 and earlier revisions: Reset: 0000 0000 0000 0020h. Revision B3 Reset: 0000 0000 1020 0020h. All defined fields are read-write.

Bits	Description
63:49	Reserved.
48	**WbEnhWsbDis: disable multi-stream write combining**. 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)

Reset: 0000 0000 0100 80C0h.

Bits	Description
63:30	Reserved.
29	**Smash1GPages**. Read-write. 1=1G pages are smashed and installed in the TLB as 2M pages.
28:16	Reserved.
15	**ClLinesToNbDis**. Read-write. 1=Clean victims/copybacks are not sent to the NB regardless of whether the L3 cache is enabled or not. 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 also section 2.3.3 [Using L2 Cache as General Storage During Boot] on page 24.
14:3	Reserved.
2	**FrcWTMemTypToWPDis**. Read-write. 1=Disable forcing of WT memory type to WP. This bit must be clear if the processor contains an L3. See section 2.3.3 [Using L2 Cache as General Storage During Boot] on page 24.
1:0	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 com-

pletes 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 also section 2.17.2 [Instruction Based Sampling (IBS)] on page 136.

Bits	Description
63:58	Reserved.
57	**IbsRandEn: random instruction fetch tagging enable**. 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	**IbsL2TlbMiss: instruction cache L2TLB miss**. Read-only. Reset X. 1=The instruction fetch missed in the L2 TLB.
55	**IbsL1TlbMiss: instruction cache L1TLB miss**. Read-only. Reset X. 1=The instruction fetch missed in the L1 TLB.
54:53	**IbsL1TlbPgSz: instruction cache L1TLB page size**. 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. Bits Definition 00b 4 Kbyte 01b 2 Mbyte 1Xb Reserved
52	**IbsPhyAddrValid: instruction fetch physical address valid**. Read-only. Reset X. 1=The physical address in MSRC001_1032 and the IbsL1TlbPgSz field are valid for the instruction fetch.
51	**IbsIcMiss: instruction cache miss**. Read-only. Reset X. 1=The instruction fetch missed in the instruction cache.
50	**IbsFetchComp: instruction fetch complete**. Read-only. 1=The instruction fetch completed and the data is available for use by the instruction decoder.
49	**IbsFetchVal: instruction fetch valid**. Read-write; set-by-hardware. 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 MSRC001_103A[LvtOffset]. This bit must be cleared and IbsFetchCnt must be written to 0000h for the fetch counter to start counting again.
48	**IbsFetchEn: instruction fetch enable**. Read-write. 1=Instruction fetch sampling is enabled.
47:32	**IbsFetchLat: instruction fetch latency**. 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	**IbsFetchCnt**. Read-write; controlled-by-hardware. Reset 0000h. This field returns the current value of bits 19:4 of the periodic fetch counter.
15:0	**IbsFetchMaxCnt**. Read-write. Reset 0000h. This field specifies maximum count value of the periodic fetch counter. Programming this field to 0000h and setting IbsFetchEn results in undefined behavior. Bits 19:4 of the maximum count are programmed in the field. Bits 3:0 of the maximum countare always 0000.

MSRC001_1031 IBS Fetch Linear Address Register (IbsFetchLinAd)

Reset: xxxx xxxx xxxx xxxxh.

Bits	Description
63:0	**IbsFetchLinAd: instruction fetch linear address**. 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 xxxxh.

Bits	Description
63:0	**IbsFetchPhysAd: instruction fetch physical address**. Read-only. This provides the physical address in canonical form for the tagged instruction fetch. The lower 12 bits are not modified by address translation, so they are always the same as the linear address. This field contains valid data only if MSRC001_1030[IbsPhyAddrValid] is asserted.

MSRC001_1033 IBS Execution Control Register (IbsOpCtl)

Reset: 0000 0000 0000 0000h.The IBS execution sampling engine tags a micro-op that 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 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 also section 2.17.2 [Instruction Based Sampling (IBS)] on page 136.

Bits	Description
63:19	Reserved.
18	**IbsOpVal: micro-op sample valid**. Read-write; set-by-hardware. 1=New instruction execution data available. When this bit is set, the periodic op counter stops counting until software clears the bit and an interrupt is generated as specified by the APIC LVT specified by MSRC001_103A[LvtOffset].
17	**IbsOpEn: micro-op sampling enable**. Read-write. 1=Instruction execution sampling enabled.
16	Reserved.
15:0	**IbsOpMaxCnt: periodic op counter maximum count**. Read-write. 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 xxxxh.

Bits	Description
63:0	**IbsOpRip: micro-op linear address**. 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	**IbsOpBrnRet: branch micro-op retired**. Reset X. Read-only. 1=Tagged operation was a branch that retired.
36	**IbsOpBrnMisp: mispredicted branch micro-op**. Read-only. Reset X. 1=Tagged operation was a branch micro-op that was mispredicted.
35	**IbsOpBrnTaken: taken branch micro-op**. Read-only. Reset X. 1=Tagged operation was a branch micro-op that was taken.
34	**IbsOpReturn: return micro-op**. Read-only. Reset X. 1=Tagged operation was return micro-op.
33	**IbsOpMispReturn: mispredicted return micro-op**. Read-only. Reset X. 1=Tagged operation was a mispredicted return micro-op.
32	**IbsOpBrnResync: resync micro-op**. Read-only. Reset X. 1=Tagged operation was resync micro-op.
31:16	**IbsTagToRetCtr: micro-op tag to retire count**. 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 IbsCompToRetCtr when the tagged micro-op is a NOP.
15:0	**IbsCompToRetCtr: micro-op completion to retire count**. 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	**NbIbsReqCacheHitSt**: **IBS L3 cache state**. Read-write; controlled-by-hardware. Valid when the data source type is Cache(2h). 0 = 'M' State. 1 = 'O' State.
4	**NbIbsReqDstProc**: **IBS request destination processor**. Read-write; controlled-by-hardware. 0=The request is serviced from the local processor. 1=The request is serviced from a remote processor.
3	Reserved.
2:0	**NbIbsReqSrc: Northbridge IBS request data source**. Read-write. 0h=No valid status. 1h=L3: data returned from local L3 cache. 2h=Cache: data returned from a CPU cache or a remote L3. 3h=DRAM: data returned from DRAM. 4h=Reserved for remote cache. 5h=Reserved. 6h=Reserved. 7h=Other: data returned from MMIO/Config/PCI/APIC.

MSRC001_1037 IBS Op Data 3 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	**IbsDcMissLat: data cache miss latency**. 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.
31:19	Reserved
18	**IbsDcPhyAddrValid: data cache physical address valid**. Read-only. Reset X. 1=The physical address in MSRC001_1039 is valid for the load or store operation.
17	**IbsDcLinAddrValid: data cache linear address valid**. Read-only. Reset X. 1=The linear address in MSRC001_1038 is valid for the load or store operation.
16	**IbsDcMabHit: MAB hit**. Read-only. Reset X. 1=The tagged load or store operation hit on an already allocated MAB. IBS data in MSRC001_1036 is not valid if this bit is set.
15	**IbsDcLockedOp: locked operation**. Read-only. Reset X. 1=Tagged load or store operation is a locked operation.
14	**IbsDcWcMemAcc: WC memory access**. Read-only. Reset X. 1=Tagged load or store operation accessed write combining memory.
13	**IbsDcUcMemAcc: UC memory access**. Read-only. Reset X. 1=Tagged load or store operation accessed uncacheable memory.
12	**IbsDcStToLdCan: data forwarding from store to load operation cancelled**. Read-only. Reset X. 1=Data forwarding from a store operation to the tagged load was cancelled.
11	**IbsDcStToLdFwd: data forwarded from store to load operation**. 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	**IbsDcStBnkCon: bank conflict on store operation**. 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	**IbsDcLdBnkCon: bank conflict on load operation**. 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	**IbsDcMisAcc: misaligned access**. Read-only. Reset X. 1=The tagged load or store operation crosses a 128 bit address boundary.
7	**IbsDcMiss: data cache miss**. Read-only. Reset X. 1=The cache line used by the tagged load or store was not present in the data cache.
6	**IbsDcL2tlbHit2M: data cache L2TLB hit in 2M page**. Read-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	**IbsDcL1tlbHit1G: data cache L1TLB hit in 1G page**. 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	**IbsDcL1tlbHit2M: data cache L1TLB hit in 2M page**. 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	**IbsDcL2tlbMiss: data cache L2TLB miss**. Read-only. Reset X.1=The physical address for the tagged load or store operation was not present in the data cache L2TLB.
2	**IbsDcL1tlbMiss: data cache L1TLB miss**. Read-only. Reset X.1=The physical address for the tagged load or store operation was not present in the data cache L1TLB.
1	**IbsStOp: store op**. Read-only. Reset X.1=Tagged operation is a store operation
0	**IbsLdOp: load op**. Read-only. Reset X.1=Tagged operation is a load operation

MSRC001_1038 IBS DC Linear Address Register (IbsDcLinAd)

Reset: xxxx xxxx xxxx xxxxh.

Bits	Description
63:0	**IbsDcLinAd**. 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 MSRC001_1037[IbsDcLinAddrValid] is asserted.

MSRC001_1039 IBS DC Physical Address Register (IbsDcPhysAd)

Reset: xxxx xxxx xxxx xxxxh.

Bits	Description
63:0	**IbsDcPhysAd**. Read-only. This provides the physical address in canonical form for the tagged load or store operation. The lower 12 bits are not modified by address translation, so they are always the same as the linear address. This field contains valid data only if MSRC001_1037[IbsDcPhyAddrValid] is asserted.

MSRC001_103A IBS Control Register

Reset: 0000 0000 0000 0000h. This register is a read-only copy of F3x1CC.

Bits	Description
63:9	Reserved.
8	**LvtOffsetVal: local vector table offset valid**. Read-only. 1=The offset in LvtOffset is valid.
7:4	Reserved.
3:0	**LvtOffset: local vector table offset**. Read-only. This specifies the address of the IBS LVT entry in the APIC registers as follows: LVT address = (LvtOffset << 4) + 500h (see APIC[530:500]).

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] on page 351.
- EventSelect 0DBh [FPU Exceptions] on page 352.

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 also Event-Select 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] on page 346.

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 CPUID Instructions

The number of CPUID 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.

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

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 Streaming Store event reflects only flushes due to streaming stores (which are typically only to WB memory). The difference between counts of these two events reflects the true amount of write events to WC memory.

UnitMask	Description
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).

UnitMask	Description
01h	Cancelled prefetches
02h	Prefetch attempts

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
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 eight 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 HALT 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 EventSelect 041h 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.

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 evicticed 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.
04h	SMC that did not hit any in-flight instructions.
08h	SMC 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] on page 342.
- EventSelect 027h [Retired CPUID Instructions] on page 342.
- EventSelect 076h [CPU Clocks not Halted] on page 347.

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 and SSE2 instructions

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 also 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 also 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 also 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 also 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.

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: DRAM_width_in_bytes * 2 edges_per_memclk * 2
R/W turnaround: DRAM_width_in_bytes * 2 edges_per_memclk * 1
R/W turnaround: DRAM_width_in_bytes * 2 edges_per_memclk * (Tcl-1)

where DRAM_width_in_bytes is 8 or 16 (for single- or dual-channel systems), and Tcl is the CAS latency of the DRAM in memory system clock cycles (where the memory clock for DDR-400, or PC3200 DIMMS, for example, would be 200 MHz).

UnitMask	Description
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 also 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)	Sub-cache-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 combin-

ing 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 do 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 do 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 do 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 unganged.

UnitMask	Description
01h	Command DWORD sent
02h	Data DWORD sent
04h	Buffer release DWORD sent
08h	Nop DW sent (idle)
10h	Address extension 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.

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

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.

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

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.

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

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

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.