BIOS and Kernel Developer’s Guide (BKDG) For AMD Family 11h Processors

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

Trademarks

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

MMX is a trademark of Intel Corporation.

Windows Vista is a registered trademark of Microsoft Corporation.

HyperTransport is a trademark of the HyperTransport Technology Consortium.

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

Table of Contents

List of Figures

List of Tables

Revision History

Revision 3.00

- Initial public release.

1 Overview

The AMD Family 11h processor (in this document referred to as *the processor*) is a processing unit that supports x86-based instruction sets. It includes (a) up to two independent central processing unit cores (referred to as *cores*), (b) one high-speed HyperTransport™ I/O Link, and (c) up to two double-data rate 2 (DDR2) system memory DRAM interfaces.

AMD Family 11h processors are distinguished by the combined ExtFamily and BaseFamily fields of the CPUID instruction (see CPUID Fn0000_0001_EAX in section 3.9 [CPUID Instruction Registers]).

1.1 Intended Audience

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

1.2 Reference Documents

- Advanced Configuration and Power Interface (ACPI) Specification. www.acpi.info.
- 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.
- Socket S1g2 Processor Functional Data Sheet, #41258.
- AMD Family 11h Processor Electrical Data Sheet, #40683.
- AMD Voltage Regulator Specification, #40182.
- HyperTransport^{TM} 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.
- Revision Guide for AMD Family 11h Processors, #41788.

1.3 Conventions

1.3.1 Numbering

- **Binary numbers**. Binary numbers are indicated by appending a “b” at the end, e.g., 0110b.
- **Decimal numbers**. Unless specified otherwise, all numbers are decimal. Note: this rule does not apply to the register mnemonics described in section 3.1 [Register Descriptions and Mnemonics]; register mnemonics all utilize hexadecimal numbering.
- **Hexadecimal numbers**. Hexadecimal numbers are indicated by appending an “h” to the end, e.g., 45F8h.
- **Underscores in numbers**. Underscores are used to break up numbers to make them more readable. They do not imply any operation. E.g., 0110_1100b.
- **Undefined digit**. An undefined digit, in any radix, is notated as a lower case “x”.

1.3.2 Arithmetic and Logical Operators

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

Table 1: Arithmetic and Logical Operators

Symbol	Definition
{}	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].
[]	Square brackets are used to indicate a range of values and can take two forms. The unit of the range is implied by context. (E.g. register bit index, register number.) 1) [y:x]: A contiguous range of values is specified, from x being the least significant to y being the most significant. 2) [z,...,y,x]: A comma separated list of values is specified, from x being the least significant to z being the most significant.
|	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

- **Altvid**. Alternate voltage ID. See 2.4.1.5 [Alternative Voltage (altvid)].
- **AP**. Application processor. See section 2.3 [Processor Initialization].
- **BCS**. Base configuration space. See section 2.11 [Configuration Space].
- **BERT**. Bit error rate tester. A piece of test equipment that generates arbitrary test patterns and checks that a device under test returns them without errors.
- **BIST**. Built-in self-test. Hardware within the processor that generates test patterns and verifies that they are stored correctly (in the case of memories) or received without error (in the case of the link).
- **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. See section 2.3 [Processor Initialization].
- **C0, C1**. These are ACPI-defined core power states. C0 is operational. C1 is when the core is in halt. See sec-

tion 2.4 [Power Management].

- **C1E**. C1 enhanced state. Power-savings mode that is employed when all cores of a CMP processor are in the halt state. See [The Interrupt Pending and CMP-Halt Register] MSRC001_0055.
- **Channel.** See DRAM channel.
- **Channel interleaved mode**. Mode in which DRAM address space is interleaved between DRAM channels. See section 2.8.7 [Memory Interleaving Modes].
- **CMP**. Chip multi-processing. Refers to processors that include multiple cores. See section 2.1 [Processor Overview].
- **COF**. Current operating frequency of a given clock domain. See section 2.4.2 [P-states].
- **Cold reset**. PWROK is deasserted and RESET_L is asserted. See section 2.3 [Processor Initialization].
- **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.
- **CLKIN.** The reference clock provided to the processor, nominally 200Mhz.
- **Core**. The instruction execution unit(s) of the processor. See section 2.1 [Processor Overview].
- **CpuCoreNum**. Specifies the core number. See section 2.9.2 [Number of Cores and Core Number].
- **CPUID function X**. Refers to the CPUID instruction when EAX is preloaded with X. See section 3.9 [CPUID Instruction Registers].
- **CS**. Chip select. See F2x[1,0][4C:40] [DRAM CS Base Address Registers].
- **DC coupled**. Refers to the method used for link termination. See section 2.7.2 [Termination and Compensation].
- **DCT.** DRAM controller. See section 2.8 [DRAM Controller (DCT)].
- **DEV**. DMA exclusion vector. See section 2.6.2 [DMA Exclusion Vectors (DEV)].
- **DID**. Divisor identifier. Specifies the post-PLL divisor used to reduce the COF. See section 2.4.2 [P-states].
- **Display refresh or DR**. Traffic used for display refresh in UMA systems. See section 2.6.3.2.1 [Display Refresh And IFCM].
- **Doubleword**. A 32-bit value.
- **Downcoring**. Removal of cores. See section 2.9.2 [Number of Cores and Core Number].
- **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 Controller (DCT)].
- **DS**. Downstream. Refers to the direction of data on a link.
- **Dual-Plane**. Refers to a processor or systemboard where VDD and VDDNB are separate and may operate at independent voltage levels. Refer to 2.4.1 [Processor Power Planes And Voltage Control].
- **DW or DWORD**. Doubleword. A 32-bit value.
- **ECS**. Extended configuration space. See section 2.11 [Configuration Space].
- **EDS**. Electrical data sheet.
- **FDS**. Functional data sheet; there is one FDS for each package type.
- **FID**. Frequency identifier. Specifies the PLL frequency multiplier for a given clock domain. See section 2.4.2 [P-states].
- **GB or Gbyte**. Gigabyte; 1,073,741,824 bytes.
- **Ganged**. A link, memory channel, or voltage regulator in which all portions are controlled as one.
- **Gen1**. Refers to older revisions of the link specification and, in particular, link data rates from 0.4 to 2.0 GT/s. Gen1 = (F0x88[Freq]<=6h). See section 2.7 [Link].
- **Gen3**. Refers to revision 3.00 of the link specification and, in particular, link data rates from 2.4 to 5.2 GT/s. Gen3 = (F0x88[Freq]>=7h). See section 2.7 [Link].
- **#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 2.10.4.1 [Hardware Thermal Control (HTC) and PROCHOT_L].
- **HTC-active state**. Hardware-controlled lower-power, lower-performance state used to reduce temperature.

See section 2.10.4.1 [Hardware Thermal Control (HTC) and PROCHOT_L].

- **HTG**. The link controller. See section 2.6 [The Northbridge (NB)].
- **I2C**. Protocol on which the SVI interface timing is based. See section 2.4.1 [Processor Power Planes And Voltage Control], and section 1.2 [Reference Documents].
- **IFCM**. Isochronous flow-control mode, as defined in the link specification.
- **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] F0x170.
- **IO configuration**. Access to configuration space through IO ports CF8h and CFCh. See section 2.11 [Configuration Space].
- **IO Hub**. This is the platform device that contains the bridge to the system BIOS.
- **IO link**. A link configured for non-coherent traffic, per the 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.
- **Link**. A link as specified by the link specification.
- Link Specification. The *HyperTransport™ I/O Link Specification* document.
- **LINT**. Local interrupt.
- **LMM**. Link Management Mode. See the link specification and [The LMM Configuration Registers] F4x174_x[0F:00] for details.
- **Logical DIMM**. Either one 64-bit DIMM or two identical DIMMs in parallel to create a 128-bit interface. See section 2.8 [DRAM Controller (DCT)].
- **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].
- **MCQ**. Memory controller queue. See section 2.6.1 [Northbridge (NB) Architecture].
- **MEMCLK**. Refers to the clock signals, M[B, A]_CLK_[H,L][7,5,4,1], that are driven from the processor to DDR DIMMs.
- **ms**. Millisecond.
- **MMIO**. Memory-mapped input-output range. This is physical address space that is mapped to the IO functions such as the IO link 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].
- **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].
- **MSR**. Model specific register. The CPU includes several MSRs for general configuration and control. See section 3.10 [MSRs - MSR0000_xxxx] for the beginning of the MSR register definitions.
- **MTRR**. Memory-type range register. The MTRRs specify the type of memory associated with various memory ranges. See MSR0000_00FE, MSR0000_02[0F:00], MSR0000_02[6F:68, 59, 58, 50], and MSR0000_02FF.
- **NB**. Northbridge. The transaction routing block of the processor. See section 2.1 [Processor Overview].
- **NCLK**. The NB clock. The NCLK frequency is the MemClk frequency.
- **Normalized address**. Addresses used by DCTs. See section 2.6.1 [Northbridge (NB) Architecture].
- **ns**. Nanosecond.

- **Octword**. A 128-bit value.
- **ODT**. On-die termination, which is applied DRAM interface signals.
- Onion. The interface that connects the NB to the HTG.
- **Operational frequency**. The frequency at which the processor operates. See section 2.4.2 [P-states].
- **PDS**. Product data sheet.
- **PRBS**. Pseudo-random bit sequence.
- **Processor**. See section 2.1 [Processor Overview].
- **P-state**. Performance state. See section 2.4.2 [P-states].
- **Pmin**. Maximum enabled P-state number (minimum power state). See section 2.4.2 [P-states].
- **PTDS**. Power and thermal data sheet.
- **PTE**. Page table entry.
- **Quadword**. A 64-bit value.
- **RAS**. Reliability, availability and serviceability (industry term). See section 2.13 [RAS: Reliability, Availability, and Serviceability].
- Revision guide. The *Revision Guide for Family 11h Processors* document.
- **RX**. Receiver.
- **S3.** ACPI-defined suspend-to-ram state. See section 2.4.4 [ACPI Suspend to RAM State (S3)].
- **SB-TSI**. SMBus-based sideband temperature sensor interface. See section 2.10.3 [Sideband Temperature Sensor Interface (SB-TSI)].
- **Scrubber**. Background memory checking logic. See section [].
- Shutdown. A state in which the affected core waits for either INIT, RESET, or NMI. When shutdown state is entered, a shutdown special cycle is sent on the IO links.
- Single-Plane. Refers to a processor or systemboard where VDD and VDDNB are tied together and operate at the same voltage level. Refer to 2.4.1 [Processor Power Planes And Voltage Control].
- **Slam**. Refers to change the voltage to a new value in one step (as opposed to stepping). See section 2.4.1.8 [Hardware-Initiated Voltage Transitions].
- **SMAF**. System management action field. This is the code passed from the SMC to the processors in STP-CLK assertion messages. The action taken by the processors in response to this message is specified by [The ACPI Power State Control Registers] F3x[84:80].
- **SMBus**. System management bus. Refers to the protocol on which the serial VID interface (SVI) commands and SBI are based. See section 2.4.1 [Processor Power Planes And Voltage Control], 2.10.3 [Sideband Temperature Sensor Interface (SB-TSI)], and section 1.2 [Reference Documents].
- **SMC**. System management controller. This is the platform device that communicates system management state information to the processor through an IO link, typically the system IO Hub.
- **SMI**. System management interrupt. See section 2.14.2.1 [SMM Overview].
- **SMM**. System management mode. See section 2.14.2 [System Management Mode (SMM)].
- **Southbridge**. Same as IO Hub.
- **Speculative event**. A performance monitor event counter that counts all occurrences of the event even if the event occurs during speculative code execution.
- **SVI**. Serial VID interface. See section 2.4.1 [Processor Power Planes And Voltage Control].
- **SVM**. secure virtual machine. See section 2.15 [Secure Virtual Machine Mode (SVM)].
- **Sync flood**. The propagation of continuous sync packets to the link. See the link specification.
- **TCB**. Trace capture buffer.
- **TSM**. Therm sense macro. See section 2.10 [Thermal Functions].
- **Tctl**. Processor temperature control value. See section 2.10.4 [Temperature-Driven Logic].
- **Thermal diode**. A diode connected to the THERMDA and THERMDC pins used for thermal measurements. See section 2.10.2 [Thermal Diode].
- **Triple-Plane**. Refers to a processor or systemboard where VDD0, VDD1, and VDDNB are separate and may operate at independent voltage levels. Refer to 2.4.1 [Processor Power Planes And Voltage Control].
- **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.
- **us**. Microsecond.
- VC. Virtual channel. See section 2.6.3.2.1 [Display Refresh And IFCM].
- **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].
- **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].
- **WDT**. Watchdog timer. A timer that detects activity and triggers an error if a specified period of time expires without the activity.
- **XBAR**. Cross bar; command packet switch. See section 2.6.1 [Northbridge (NB) Architecture].

1.5 Changes Between Revisions and Product Variations

1.5.1 Major Changes For Revision A Relative to Family 0Fh Processors

- Core(s):
  - Supports the same functionality as the Family 0Fh revision G core.
  - Support for up to 2 cores in product variations.
  - Power management state invariant time stamp counter (TSC).
  - All local sources of SMIs (including sources from the cores and from the NB) are broadcast to all cores in the processor.
- Memory controller (DCT):
  - Support for DDR2 DIMMs in product variations.
  - Write burst and DRAM prefetching performance improvements.
- Link and IO additions:
  - Gen3 link specification, including support for DC-coupled mode. (No support for AC-coupled mode.)
  - 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.
- General NB:
  - Ability for core(s) and memory controller to operate when the link is disconnected. See LDTREQ_L and Centralized Link Power Management Control in the link specification.
  - BIOS-initiated system memory clear command.
  - MMIO-based access to configuration space and support for extended configuration space.
  - Mode whereby the IO request response order matches the IO request order.
  - VGA space decoding to MMIO-space mapping registers.
  - More DEV protection domains and a larger DEV cache.
  - Added support for Source ID extension packets to support finer grain control of DEV protection domains.
- Power management:
  - Simple “fire and forget” operating system interface for P-state changes.
  - Gen3 link power management. (CDLW, CDLF, CDLC, CDLD)
  - Separate core and NB power and clock planes.
  - Support for up to 8 independent P-states for each core.

- Support for P-state limits controlled by sideband interface (SB-TSI), thermal limits (HTC), or host software; used to limit the P-state requested by the operating system in order to reduce power.
- Support for altvid exit in Pmin (MSRC001_0061[PstateMaxVal]) state to service probes.

2 Functional Description

2.1 Processor Overview

The processor is an integrated circuit device that includes (1) one to two cores, (2) one IO link for general-purpose communication to other devices, (3) one or two 64-bit DDR2 DRAM interfaces for communication to system memory, and (4) one communication packet routing block referred to as the *northbridge* (NB).

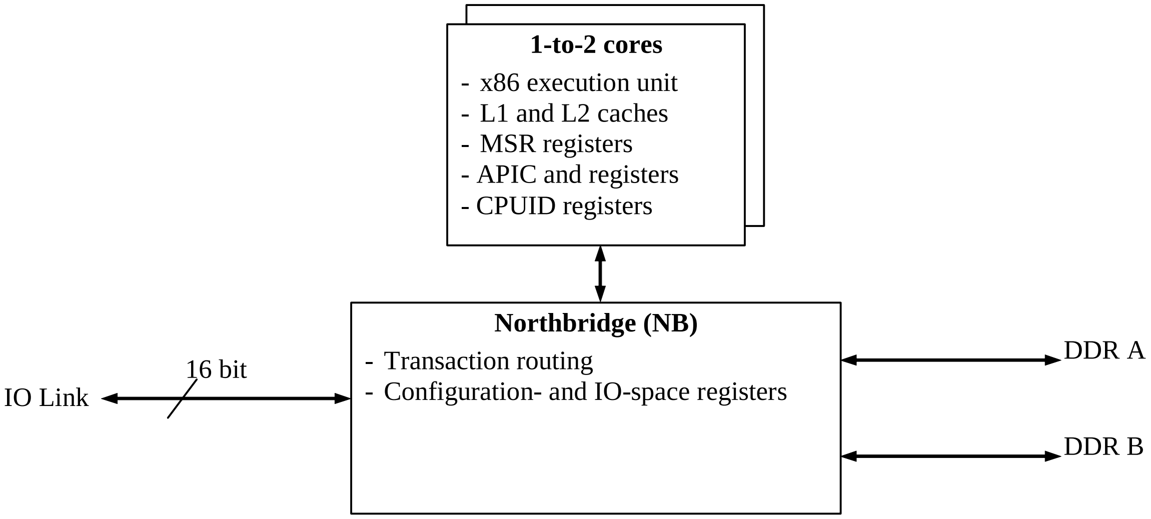

Figure 1: A processor

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. Processors that include multiple cores are said to incorporate *chip multi-processing* or CMP.

The IO link is an input-output link, as defined by the link specification.

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

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

2.2 System Overview

The following diagram illustrates the expected system architecture:

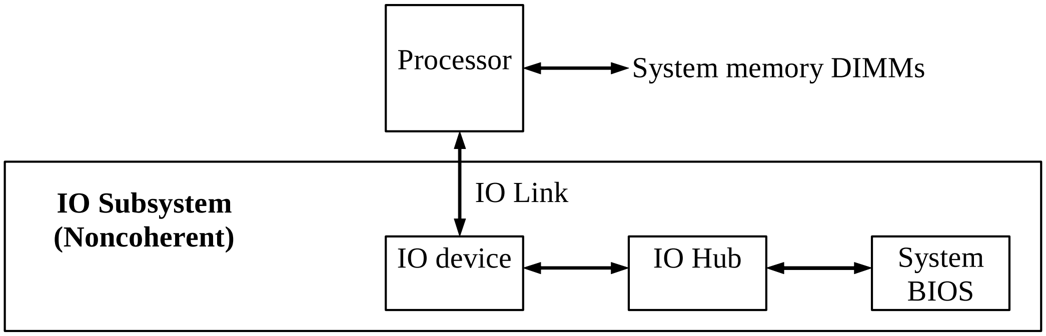

Figure 2: System Diagram

2.3 Processor Initialization

This section describes the initialization sequence after a cold reset.

The BSC, core 0, begins executing code from the reset vector. The AP core, core 1, does not fetch code until its enable bit, F0x68 [Cpu1En], is set.

2.3.1 BSC initialization

The BSC 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). 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]).
- Enable the cache, program the MTRRs for Cache-as-Ram and initialize the Cache-as-Ram, as described in 2.3.3 [Using L2 Cache as General Storage During Boot].
- Setup of APIC (2.9.5.1 [APIC ID Enumeration Requirements]).
- Configure all IO-link devices.
- Device enumeration for all IO-link devices (see link specification).
- If required, reallocate data and flow control buffers of the link (see [The Link Base Channel Buffer Count Register] F0x90 and [The Link Isochronous Channel Buffer Count Register] F0x94) and issue system warm reset.
- Configure the link speed and link width (see link specification).
- Configure processor power management (see 2.4 [Power Management]).
- If supported, allow the other core to beginning fetching instructions by setting [The Link Transaction Control Register] F0x68[Cpu1En].

2.3.2 AP initialization

The other processor core (core 1) begins executing code from the reset vector. It 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 the 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). 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]).
- Setup of local APIC (2.9.5.1 [APIC ID Enumeration Requirements]).
- Configure processor power management (see 2.4 [Power Management]).

2.3.3 Using L2 Cache as General Storage During Boot

Prior to initializing the DRAM controller for system memory, BIOS may use the L2 cache of each core as 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 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 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:
        - Picks an invalid way before picking a valid way.
        - Prioritizes the picking of invalid ways such that way 0 is the highest priority and L2WayNum-1 is the lowest priority.
- 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.
  - All of memory space that is not accessed as WB-DRAM space must be marked as UC memory type.

Performance monitor event EventSelect 07Fh [L2 Fill/Writeback], subevent 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_CLR_WBTOL2_SMC_HIT]=1.
- MSRC001_1022[DIS_HW_PF]=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_CLR_WBTOL2_SMC_HIT]=0, MSRC001_1022[DIS_HW_PF]=0, MSRC001_0015[INVD_WBINVD].

2.3.4 BIOS Requirements For 64-Bit Operation

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

2.4 Power Management

The processor supports many power management features in a variety of systems. Table 2 provides a summary of ACPI states and power management features and indicates whether they are supported.

Table 2: Power management support

ACPI/Power Management State	Supported	Description
G0/S0/C0: Working	Yes	
G0/S0/C0: Core P-state transitions	Yes	2.4.2 [P-states]
G0/S0/C0: NB P-state transitions	No	
G0/S0/C0: Hardware thermal control (HTC)	Yes	2.10.4.1 [Hardware Thermal Control (HTC) and PROCHOT_L]

Table 2: Power management support

ACPI/Power Management State	Supported	Description
G0/S0/C0: Thermal clock throttling (SMC controlled)	No	
G0/S0/C1: Halt	Yes	
G0/S0/C2: Stop-grant Caches snoopable (single-core devices only)	No	
G0/S0/C3: Stop-grant Caches not snoopable (single-core devices only)	No	
G0/S0/C1E: Stop-grant Caches not snoopable	Yes	2.4.3.1 [C1 Enhanced State (C1E)]
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)]
G1/S4, S5: Hibernate (Suspend to Disk), Shut Down (Soft Off)	Yes	
G3 Mechanical Off	Yes	
Link Power Management	Yes	2.4.6 [Link Power States]
Parallel VID Interface	No	2.4.1 [Processor Power Planes And Voltage Control]
Serial VID Interface	Yes	
Single-plane systems	No	
Dual-plane systems	Yes	
Triple-plane systems	Yes	

2.4.1 Processor Power Planes And Voltage Control

The processor includes the following power planes:

- VDDIO: used for the DRAM and miscellaneous pins.
  - This plane is powered during S3 (see section 2.4.4 [ACPI Suspend to RAM State (S3)]).
- VTT: used for the DDR DRAM interface.
  - This plane is powered during S3 (see section 2.4.4 [ACPI Suspend to RAM State (S3)]).
  - The voltage level is specified to be half of the VDDIO level.
- VLDT: used for the link.
- VDDA: filtered PLL supply.
- VDD or VDD[1:0]: main supply for core logic.
  - VDD refers generically to the core voltage plane(s).
  - VDD0 refers to the core 0 power plane.
  - VDD1 refers to the core 1 power plane.
  - 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[1:0] and VDDNB may be altered in various states. All the other supplies are fixed. Refer to the AMD Family 11h Processor Electrical Data Sheet, #40683 for power plane sequencing requirements.

The processor supports:

- Dual-plane platforms. VDD[1:0] are tied together on the systemboard and controlled as a single voltage

plane. VDDNB is isolated from VDD[1:0] on the systemboard and is controlled as a separate voltage plane. See F3xA0[VddCpuGanged].
- Triple-plane platforms. VDD0, VDD1, and VDDNB are all isolated on the systemboard and are all controlled as independent voltage planes. See F3xA0[VddCpuGanged].

2.4.1.1 Internal VID Registers

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

- The VID for VDDNB is dictated by F3xDC[NbVid].
- If F3xA0[VddCpuGanged]=1, indicating that the processor is configured for a dual-plane system, the VID for VDD is dictated by MSRC001_00[6B:64][CpuVid] of the CPU-core in the highest-performance P-state.
- If F3xA0[VddCpuGanged]=0, indicating that the processor is configured for a triple-plane system, the VID for VDD is dictated by MSRC001_00[6B:64][CpuVid].

2.4.1.2 Serial VID Interface

The processor includes an interface, intended to control external voltage regulators, called the serial VID interface (SVI). 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 SVD, the data pin, on the rising edge. The frequency of the SVC is controlled by F3xA0[SviHighFreqSel]. The SVI protocol is specified in the AMD Voltage Regulator Specification, #40182.

2.4.1.3 MinVid and MaxVid Check

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

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

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

2.4.1.4 PSI_L Bit

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

2.4.1.5 Alternative Voltage (altvid)

In order to save power, a lower alternative voltage (altvid) may be applied while in either the C1E state, or by LMM. Altvids are controlled by F3x[84:80][CpuAltVidEn], F3xDC[AltVid], F4x174_x[0F:00][LmmCpuAltVidEn].

2.4.1.6 VID Encodings

The VID encoding to VDD translations, for both the boot VID and the SVI VID, are defined by the AMD Voltage Regulator Specification, #40182.

The boot VID is 1.1 volts.

2.4.1.7 BIOS Requirements for Power Plane Initialization

- BIOS must initialize F3xA0[VddCpuGanged].
- Optionally configure F3xA0[PsiVidEn and PsiVid]. Refer to section 2.4.1.4 [PSI_L Bit] for additional details.
- BIOS must initialize F3xD8[VSSlamTime].

2.4.1.8 Hardware-Initiated Voltage Transitions

VDD and VDDNB voltage levels may be transitioned during state changes involving boot, reset, P-state, and stop-grant. In all cases, the voltage is *slammed*; this means that the VID code passed to the voltage regulator changes from the old value to the new value without stepping through intermediate values. The voltage regulator ramps the voltage directly from the starting voltage to the final voltage, no stepping occurs. See the AMD Voltage Regulator Specification, #40182 and F3xD8[VSSlamTime] for details.

2.4.1.9 Software-Initiated Voltage Transitions

The processor supports direct software VID control using [The COFVID Control Register] MSRC001_0070. Hardware P-state transitions using [The P-state Control Register] MSRC001_0062 result in unpredictable behavior if software modifies the CpuVid from the appropriate settings for the current P-state reported in [The P-state Status Register] MSRC001_0063.

1. Write the destination CPU VID to MSRC001_0070[CpuVid].
2. Wait the specified F3xD8[VSSlamTime].

2.4.2 P-states

P-states are operational performance states characterized by a unique combination of frequency and voltage. The processor supports dynamic P-state changes for up to two frequency domains and up to two voltage planes: VDD0 and VDD1. Refer to section 2.4.1 [Processor Power Planes And Voltage Control] for voltage plane definitions. Up to 8 core P-states, called P0 though P7, are supported. P0 is the highest-power, highest-performance P-state; each ascending P-state number represents a lower-power, lower-performance P-state. Lower performance P-states (higher numbered P-states) must have a COF that is less than or equal to higher performance P-states (lower numbered P-states). Lower performance P-states must also have a VID that is higher than or equal to (voltage that is lower than or equal to) higher performance P-states. The number of defined P-states is optimized for power and performance trade-offs. At least one P-state (P0) is enabled and specified for all processors. Out of cold reset, the voltage and frequency of the cores is specified by MSRC001_0071[StartupPstate].

The following terminology applies to P-state definitions:

- FID: frequency ID. Specifies the PLL frequency multiplier, relative to the reference clock, for a given domain.
- DID: divisor ID. Specifies the post-PLL power-of-two divisor that may be used to reduce the operating frequency.

- COF: current operating frequency.
  - Refer to 2.4.2.1 [Core P-states] for details on the reference clock frequency and allowed DIDs for core P-states.
  - Refer to MSRC001_00[6B:64][CpuFid] for the CPU COF formula and details on allowed FIDs for core P-states.
- 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[6B:64]) after a cold reset.
- VID: voltage ID. Specifies the voltage level for a given domain.

2.4.2.1 Core P-states

If [The P-state [7:0] Registers] MSRC001_00[6B:64][PstateEn] is set in more than one register, the processor supports dynamic P-state changes. The FID, DID, and VID for each core P-state is specified in [The P-state [7:0] Registers] MSRC001_00[6B:64]. 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). P-state transitions using the hardware P-state control mechanism are not allowed until the P-state initialization requirements defined in section 2.4.2.4 [BIOS Requirements for P-state Initialization and Transitions] are complete.

Refer to the MSRC001_00[6B:64] and definition for further details on programming requirements.

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 2.4.2.6 [ACPI Processor P-state Objects]). Software requests a core P-state change by writing a 3-bit index corresponding to the desired core P-state number to [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 [7:0] Registers] MSRC001_00[6B: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 2.4.2.3 [P-state Transition Behavior] with no additional software interaction required. [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 2.4.2.3 [P-state Transition Behavior] for details on hardware P-state voltage control. Section 2.4.1.7 [BIOS Requirements for Power Plane Initialization] specifies the processor initialization requirements for voltage plane control.

Core P-states are changed without interaction with the IO Hub. However, the IO Hub is notified of core P-state changes by the P-state special cycle if MSRC001_001F[EnaPStateSpCyc]=1.

2.4.2.2 P-state Limits

P-states may be also limited to lower-performance values under certain conditions, including HTC. HTC is controlled by [The Hardware Thermal Control (HTC) Register] F3x64[HtcPstateLimit]. 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 P-state value, regardless of the source, is limited as specified in MSRC001_0061[PstateMaxVal].

2.4.2.3 P-state Transition Behavior

P-state changes normally include a COF change and may include a VID change. If the P-state number is increasing (the core is moving to a lower-performance state), then the COF is changed first, followed by the VID change. If the P-state number is decreasing, then the VID is changed first followed by the COF.

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 voltage domain, it completes 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.

Each code has one set of P-state control registers. Each core may independently request to enter a different P-state. However, depending on the configuration of power planes specified by F3xA0[VddCpuGanged], the COF and the VID associated with each P-state may not be in sync. 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 that two cores are both initially in P0 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. If multiple P-state changes are requested concurrently, the PSSM may group the associated VID changes separately from the associated COF changes.
- If RESET_L asserts the processor cores are transitioned to C0 and to the power state specified by MSRC001_0071.
  - After a warm reset [The P-state Control Register] MSRC001_0062 and [The P-state Status Register] MSRC001_0063 are consistent with MSRC001_0071[CurPstate].
  - After a warm reset MSRC001_0070 may not reflect MSRC001_0071. See [The BIOS COF and VID Requirements After Reset] 2.4.2.9.
- If an altvid is requested during a P-state transition:
  - The P-state transition is allowed to complete before the altvid is applied.
  - When exiting the altvid state, the processor places the VID at it's last value prior to the altvid.
- If a probe needs to be serviced while in an altvid low power state, a transition is made to a higher performance P-state based on F3x[84:80][CpuPrbEn].
- The OS controls the P-state through [The P-state Control Register] MSRC001_0062, independent of the P-state limits described in MSRC001_0061[PstateMaxVal, CurPstateLimit]. P-state limits interact with OS-

directed P-state transitions as follows:

- MSRC001_0061[PstateMaxVal] is the lowest-performance P-state limit and is treated as an lower limit on performance. See MSRC001_0061[PstateMaxVal] for the conditions that affect the lower performance P-state limit.
- MSRC001_0061[CurPstateLimit] is the highest-performance P-state limit and is treated as an upper limit on performance. See MSRC001_0061[CurPstateLimit] for the conditions that affect the upper performance P-state limit.
    - The P-state for a core is changed by hardware to MSRC001_0061[CurPstateLimit] if MSRC001_0061[CurPstateLimit] changes and the current P-state is lower than MSRC001_0061[CurPstateLimit].

- Altvid value specified by F3xDC[AltVid] may be applied in the C1E state. While in this state, the altvid value is applied; when exiting these states, the VIDs associated with the current P-state of each power plane is applied.

2.4.2.4 BIOS Requirements for P-state Initialization and Transitions

BIOS requirements are:

1. Check that CPUID Fn8000_0007[HwPstate]=1. If not, P-states are not supported, no P-state related ACPI objects should be generated, and BIOS must skip the rest of these steps.
2. Complete the requirements in 2.4.1.7 [BIOS Requirements for Power Plane Initialization].
3. Complete the requirements in 2.4.2.9 [BIOS COF and VID Requirements After Reset].
4. Determine the valid set of P-states based on the enabled P-states indicated in [The P-state [7:0] Registers] MSRC001_00[6B:64][PstateEn].
5. If only one P-state is enabled in [The P-state [7:0] Registers] MSRC001_00[6B:64][PstateEn], then BIOS must not generate ACPI-defined P-state objects described in section 2.4.2.6 [ACPI Processor P-state Objects]. Otherwise, the ACPI objects should be generated to enable P-state support.

2.4.2.5 Processor-Systemboard Power Delivery Compatibility Check

BIOS may disable processor P-states that require higher power delivery than the systemboard can support. This power delivery compatibility check is designed to prevent system failures caused by exceeding the power delivery capability of the systemboard for the power plane(s) that contain the core(s). Refer to section 2.4.1 [Processor Power Planes And Voltage Control] for power plane definitions and configuration information. BIOS can optionally notify the user if P-states are detected that exceed the systemboard power delivery capability. Modifications to [The P-state [7:0] Registers] MSRC001_00[6B:64] must be applied equally to all cores. This check does not ensure functionality for all package/socket compatible processor/systemboard combinations.

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

ProcIddMax = MSRC001_00[6B:64][IddValue] * 1/10^MSRC001_00[6B:64][IddDiv] * (F3xE8[CmpCap]+1);

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.

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.

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

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

1. 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[6B:64]) to the highest performance P-state locations. E.g. if P0 and P1 are disabled by the power delivery compatibility check and P2 - P4 remain enabled, then the contents of P2 - P4 should be copied to P0 - P2 and P3 and P4 should be disabled (PstateEn=0).
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]
   - F3xDC[PstateMaxVal]

If the processor has all P-states disabled after performing the power delivery compatibility check, then BIOS must perform the following steps. This does not ensure operation, and 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]
   - F3xDC[PstateMaxVal]

2.4.2.6 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 described in the following sections.

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

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

2.4.2.6.1 _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

2.4.2.6.2 _PSS (Performance Supported States)

A unique _PSS entry is created for each P-state. BIOS must loop through each of [The P-state [7:0] Registers] MSRC001_00[6B:64] applying the formulas for CoreFreq and Power, and assigning Control and Status appropriately for valid P-states (PstateEn=1). The TransitionLatency and BusMasterLatency values can be set to 0 for all _PSS entries.

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[6B:64]) that contains the COF and VID settings for the associated P-state. The value contained in [The P-state Status Register] MSRC001_0063 can be used to identify the _PSS object of the current P-state by equating MSRC001_0063[CurPstate] to the value of the Status field. Refer to section 2.4.2 [P-states] for further details on P-state definition and behavior.

- CoreFreq (MHz) = The CPU COF specified by MSRC001_00[6B:64][CpuFid] rounded to the nearest 100 Mhz.
- Power (mW)
  - Convert MSRC001_00[6B:64][CpuVid] to a voltage by referring to the AMD Voltage Regulator Specification, #40182.
  - Power (mW) = voltage * MSRC001_00[6B:64][IddValue] * 1/10^MSRC001_00[6B:64][IddDiv] * 1000
- TransitionLatency (us) = BusMasterLatency (us) = 0 us.
- Control:
  - 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
  - If MSRC001_0069 (P5): Control = 0000_0005h

  - If MSRC001_006A (P6): Control = 0000_0006h
  - If MSRC001_006B (P7): Control = 0000_0007h
- 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
  - If MSRC001_0069 (P5): Status = 0000_0005h
  - If MSRC001_006A (P6): Status = 0000_0006h
  - If MSRC001_006B (P7): Status = 0000_0007h

2.4.2.6.3 _PPC (Performance Present Capabilities)

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

2.4.2.6.4 _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.

2.4.2.6.5 Fixed ACPI Description Table (FADT) Entries

BIOS must declare the following FADT entries as 0:

- PSTATE_CNT = 00h
- CST_CNT = 00h

2.4.2.7 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.

2.4.2.8 Optimized Operating System Power Management Settings

Some operating systems provide settings that can be customized for power management algorithms. These settings allow users to optimize the algorithms for their specific hardware configurations and power/performance requirements. These settings can bias the processor toward increased performance, increased battery life, or a combination of both. The following recommendations provide optimal power and performance settings based on laboratory experiments and studies. AMD recommends OEM vendors use these settings to optimize their power management policies.

2.4.2.8.1 Windows Vista® Power Management Settings

The recommended Windows Vista® power management settings are as follows:

- Time Check = 10ms.
  - This parameter specifies the interval at which the OS considers frequency changes.
- Increase Time = 10000us.
  - This parameter specifies the time it takes to reach the maximum frequency once an application is started. In order to take advantage of the recommended 10ms Time Check value, Increase Time should be set to 10ms as well.
- Decrease Time = 10000us.
  - This parameter specifies the amount of time it takes for an idle core to reduce it's frequency. In order to take advantage of the lower Time Check value, Decrease Time should be set to 10ms as well.
- Increase Policy = Rocket.
  - This parameter specifies how fast P-states are increased. Rocket indicates a direct transition to the highest performance P-state.
- Decrease Policy = Single.
  - This parameter specifies how fast P-states are decreased. Single indicates the current P-state decreases by one on each Time Check interval.

2.4.2.9 BIOS COF and VID Requirements After Reset

Warm reset is asynchronous and can leave MSRC001_0070 in an unknown state. Since BIOS cannot always determine whether a reset was a warm or cold reset, the following requirement must be done after all resets.

- Write MSRC001_0070[CpuFid, CpuDid, CpuVid, PstateId] with MSRC001_0071[CurCpuFid, CurCpuDid, CurCpuVid, CurPstate].

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 IO Hub 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 IO Hub.

General requirements for C1E:

- The ACPI-defined C2 and C3 states must not be declared to the operating system.
  - The ACPI-defined P_LVL2_LAT should be greater than 100.
  - The ACPI-defined P_LVL3_LAT should be greater than 1000.
- C1E should only be enabled when the platform is in ACPI power management mode.
- C1E is not supported when CLMC is used by the IO Hub.

2.4.3.1.1 IO Hub Initiated C1E

When C1E is enabled and the processor detects that all cores have entered the halt state, the processor sends an IO read to the IO Hub. The IO read is directed to the ACPI-defined P_LVL3 register. This places all cores into the C1E state.

2.4.3.1.1.1 BIOS Requirements to Initialize IO Hub Initiated C1E

BIOS requirements are:

- Ensure all requirements in 2.4.3.1 [C1 Enhanced State (C1E)] are met.
- Configure the following registers on all cores:
  - Single core processors (CPUID Fn8000_0001_ECX[CmpLegacy]=0):
    - MSRC001_0055[C1eOnCmpHalt] = 0.
    - MSRC001_0055[SmiOnCmpHalt] = 1.
  - Dual core processors (CPUID Fn8000_0001_ECX[CmpLegacy]=1):
    - MSRC001_0055[C1eOnCmpHalt] = 1.
    - MSRC001_0055[SmiOnCmpHalt] = 0.
  - MSRC001_0055[IntPndMsg]=0.
  - MSRC001_0055[IORd] = 1.
  - MSRC001_0055[IOMsgAddr] = Address of the ACPI-defined P_LVL3 register.

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.

See [The DRAM Configuration Low Register] F2x[1,0]90[ExitSelfRef]. The following sequence must be performed to bring memory out of self-refresh mode:

1. Restore [The DRAM Controller Select Low Register] F2x110.
2. Restore the following registers

   Notes: BIOS restores DCT0 registers first, then DCT1 registers.
   If channel B contains DIMMs, then BIOS restores DCT phy registers, F2x[1,0]98, F2x[1,0]9C, and all the associated indexed registers (F2x[1,0]9C_xXX) for both DCT0 and DCT1.
   - [The DRAM Base/Limit Registers] F1x[44,40]
   - [The DRAM Hole Address Register] F1xF0
   - [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][4C:40]
   - [The DRAM CS Mask Registers] F2x[1,0][64:60]
   - [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 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 Output Driver Compensation Control Register] F2x[1,0]9C_x00
   - [The DRAM Write Data Timing [High:Low] Registers] F2x[1,0]9C_x[02:01]
   - [The DRAM Address/Command Timing Control Register] F2x[1,0]9C_x04
   - [The DRAM Read DQS Timing Control [High:Low] Registers] F2x[1,0]9C_x[06:05]
   - [The DRAM DQS Receiver Enable Timing Control Registers] F2x[1,0]9C_x[2B:10]
   - Restore [The DRAM Configuration High Register] F2x[1,0]94
   - [The DRAM Controller Miscellaneous Register] F2x[1,0]A0
   - [The DRAM Controller Temperature Throttle Register] F2xA4
4. Program F2x[1,0]94[MemClkFreqVal] = 1.
5. Program F2x[1,0]90[ExitSelfRef] = 1.
6. Wait until F2x[1,0]90[ExitSelfRef] = 0.
7. The memory subsystem is ready for use.

Many of the systemboard power planes for the processor are powered down during S3. Refer to section 2.4.1 [Processor Power Planes And Voltage Control] for power plane descriptions. Refer to the AMD Family 11h Processor Electrical Data Sheet, #40683 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 bit, THERMTRIP_L, etc.) during S3. Refer to the link specification for signal sequencing requirements for PWROK, RESET_L, and LDTSTOP_L during S3 entry and exit, and system management message sequencing for S3 entry and exit.

2.4.5 Centralized Link Power Management

The objective of centralized link power management is the dynamic adjustment of the link configuration. Link power management features are controlled by a centralized link management controller (CLMC). The register structures required for CLMC operation need to be established by the platform BIOS according to the link specification. The processor provides the necessary infrastructure and control features to support the following centralized link power management features:

- Link control features:
  - Centralized dynamic link configuration (CDLC) allows dynamically configure device specific link features. See 2.4.6 [Link Power States] and [The LMM Configuration Registers] F4x174_x[0F:00] for override control of link configuration.
  - Centralized dynamic link disconnection (CDLD) allows to disconnect the link.
  - Centralized dynamic link width (CDLW) allows to dynamically control the link width .
  - Centralized dynamic link frequency (CDLF) allows to dynamically control the link frequency.

- Link refresh features:
  - Centralized disconnected link refresh (CDLR) allows the CLMC to refresh DLL and phase recovery lock of a disconnected link to reduce reconnect latency.
  - Centralized inactive lane refresh (CILR) allows the CLMC to refresh DLL and phase recovery lock for inactive lanes without affecting operational lanes.

The functional description, software initialization requirements and considerations for combining CLMC features can be found in the link specification.

The following register fields must not get modified when any of the CLMC control features are enabled:

- F0x88[Freq]
- F0x84[WidthOut, WidthIn]
- F0x16C[T0Time]
- F0x170[TxInLnSt, RxInLnSt, ScrambleEn, TxLSSel, RxLSSel]

2.4.6 Link Power States

The LMM Configuration Registers (F4x174_x[0F:00]) provide override controls for various register settings that affect power consumption. A particular set of overrides is selected based on LMAF[3:0] of the link management system management. See [The LMM Configuration Registers] F4x174_x[0F:00].

2.4.6.1 SMAF and LMAF Interaction

Hardware behaves as follows:

- For C1, F3xD4[Smaf7Dis] selects between SMAF C1 and the LMM registers (F4x174_x[0F:00][20:14]).
- SMAF 0 to 6 (not C1) dominates over LMM, until a subsequent STPCLK deassertion. (e.g. S1 exit.)

When LMAF is enabled:

- F3xD4[Smaf7Dis]=1.
- SMAF 0 to 6 must only be used for S states.

2.5 Processor State Transition Sequences

2.5.1 ACPI Power State Transitions

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

The following describes the state transition behavior associated with ACPI Power State Transitions:

- All SMAF controllable parameters take effect after an LDTSTOP assertion except CpuDid, which takes affect before the first LDTSTOP assertion.
- The ACPI C1 state is controlled by F3xD4[Smaf7Dis].
- CpuDid:
  - F3x[84:80][CpuDid] is applied after the processor has transitioned from C0 to a low-power state (halt or stop-grant) and the hysteresis time (F3xD4[ClkRampHystSel]) has elapsed.
- Altvid:
  - Altvid is applied when all of the following are true:
    - CpuDid has been applied to all cores.
    - SMAF indicates altvid can be applied. (F3x[84:80][CpuAltVidEn]=1)
    - Completed all outstanding probes.
    - LDTSTOP is asserted.
  - Altvid is removed when all of the following are true:
    - LDTSTOP is deasserted.
    - At least 1 probe or an interrupt is pending.
  - When altvid is removed the next state:
    - for an interrupt and 0 or more probes is the current P-state.

        - for at least 1 probe and no interrupts is specified by F3x[84:80][CpuPrbEn, CpuAltVidEn].
- Probes
    - F3x[84:80][CpuAltVidEn, CpuPrbEn] specifies how probes are handled while in the low-power state with the following exceptions:
        - If CpuAltVidEn=1, CpuPrbEn=1 and a probe is received prior to the first LDTSTOP assertion, then the probe will be handled as if CpuAltVidEn=0 and CpuPrbEn=1.
        - If CpuAltVidEn=1 at the time that LDTSTOP asserts, then probes serviced when LDTSTOP deasserts will be handled according to the CpuPrbEn latched at the time that LDTSTOP asserted.
- DRAM:
    - DRAM self refresh is enabled if all of the following are true:
        - SMAF indicates DRAM self refresh can be applied. (F3x[84:80][DramSr]=1).
        - LDTSTOP is asserted.
    - DRAM memory clock is tristated if all of the following are true:
        - SMAF indicates DRAM memory clock tristated can be applied. (F3x[84:80][DramMemClkTri]=1).
        - LDTSTOP is asserted.
    - When LDTSTOP is deasserted the DRAM memory clock is enabled and self refresh is disabled.
        - Occurs in parallel to re-connecting the link.

2.5.2 Link Power State Transitions

This section specifies link controlled power state transitions in the APCI C1 state, as controlled by the [The LMM Configuration Registers] F4x174_x[0F:00] when F3xD4[Smaf7Dis]=1.

The following describes the state transition behavior associated with Link Power State Transitions:
- A warm reset clears the overrides and returns the link to the programmed state, as if an LMAF of 0 had been received, followed by an LDTSTOP assertion.
- All F4x174_x[0F:00] parameters take effect after an LDTSTOP assertion except LmmCpuDid, which takes affect before the first LDTSTOP assertion.
    - The link related fields take effect when LDPSTOP asserts, ignoring the state of F0x16C[ImmUpdate]; these fields are: F4x174_x[0F:00][30:21,12:0].
- CpuDid:
    - F4x174_x[0F:00][LmmCpuDid] is applied after the processor has transitioned from C0 to a low-power state (halt or stop-grant) and the hysteresis time (F3xD4[ClkRampHystSel]) has elapsed.
- Altvid:
    - Altvid is applied when all of the following are true:
        - CpuDid has been applied to all cores.
        - LMAF indicates altvid can be applied. (F4x174_x[0F:00][LmmCpuAltVidEn]=1)
        - Completed all outstanding probes.
        - LDTSTOP is asserted.
    - Altvid is removed when all of the following are true:
        - LDTSTOP is deasserted.
        - At least 1 probe or an interrupt is pending.
        - LMAF indicates altvid can't be applied. (F4x174_x[0F:00][LmmCpuAltVidEn]=0)
    - When altvid is removed the next state:
        - for an interrupt and 0 or more probes is the current P-state.
        - for at least 1 probe and no interrupts is specified by F4x174_x[0F:00][LmmCpuPrbEn, LmmCpuAltVidEn].
- Probes
    - F4x174_x[0F:00][LmmCpuAltVidEn, LmmCpuPrbEn] specifies how probes are handled while in the low-power state with the following exceptions:
        - If LmmCpuAltVidEn=1, LmmCpuPrbEn=1 and a probe is received prior to the first LDTSTOP

assertion, then the probe will be handled as if LmmCpuAltVidEn=0 and LmmCpuPrbEn=1.
- If LmmCpuAltVidEn=1 at the time that LDTSTOP asserts, then probes serviced when LDTSTOP deasserts will be handled according to the LmmCpuPrbEn latched at the time that LDTSTOP asserted. An LMAF that selects a different state for LmmCpuPrbEn will not take affect until the next LDTSTOP deassertion. An LMAF that selects LmmCpuAltVidEn=0 will cause LmmCpuAltVidEn and LmmCpuPrbEn to take affect immediately.

- DRAM:
  - DRAM self refresh is applied if all of the following are true:
    - SMAF indicates DRAM self refresh can be applied. (F4x174_x[0F:00][LmmDramSr]=1).
    - LDTSTOP is asserted.
  - DRAM memory clock tristated is applied if all of the following are true:
    - SMAF indicates DRAM memory clock tristated can be applied. (F4x174_x[0F:00][LmmDram-MemClkTri]=1).
    - LDTSTOP is asserted.
  - DRAM self refresh and memory clock tristated are exited when LDTSTOP is deasserted.
    - Exiting DRAM self refresh and memory clock tristated occurs in parallel to re-connecting the link.
- Cross-platform dependencies:
  - The NB PLL frequency can be changed without involving the link.

2.6 The Northbridge (NB)

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

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

2.6.1 Northbridge (NB) Architecture

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

The DCT operates on physical addresses and converts physical addresses into *normalized* addresses that correspond to the values programmed into [The DRAM CS Base Address Registers] F2x[1,0][4C:40]. Normalized addresses include only address bits within the DCT's 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.

The NB frequency is always equal to the actual DRAM frequency. See [The DRAM Frequency] 2.8.10.

2.6.2 DMA Exclusion Vectors (DEV)

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

2.6.3 Northbridge Routing

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

2.6.3.1 Address Space Routing

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

2.6.3.1.1 DRAM and MMIO Memory Space

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

- IO-device accesses are processed as follows:
  - If the access matches [The Memory Mapped IO Base/Limit Registers] F1x[BC:80], then the transaction is routed to the link;
  - Else, if the access matches [The DRAM Base/Limit Registers] F1x[44,40], then the access is routed to the DCT;
  - Else, the access is routed to the link which by default contains compatibility (subtractive) address space.
- For core accesses 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[44,40], then the transaction is routed to the DCT;
    - Else, the access is routed to the link which by default contains the compatibility (subtractive) address space.
  - 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 link;
    - Else, the access is routed to the link which by default contains the compatibility (subtractive) address space.

2.6.3.1.2 IO Space

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

- If the access matches [The IO-Space Base/Limit Registers] F1x[C4,C0], then the transaction is routed to the link;
- Else, the access is routed to the link which by definition contains the compatibility (subtractive) address space.

2.6.3.1.3 Configuration Space

See "Configuration Space" on page 75.

2.6.3.2 Link Routing

There are three types of link transactions routed by the NB: (1) broadcast transactions, (2) request transactions, and (3) response transactions.

2.6.3.2.1 Display Refresh And IFCM

See “Buffer Allocation and Programming Rules” on page 40 for how to allocate buffering for link requests and programming guidelines.

See “Buffer Allocation Recommendations” on page 40 for the recommendations to produce optimal performance.

Three types of upstream requests are supported: base, display refresh, and isochronous.

- Display refresh:
    - Display refresh requests are generated by UMA graphics chipsets. It targets system memory for the purpose of refreshing the display.
    - A display refresh request is defined as a non-posted read request with the isochronous bit, PassPW bit, and RespPassPW bit set, and the coherent bit cleared. The SeqID must be zero.
- Isochronous:
    - An isochronous request is defined as a request with the isochronous bit set that is not a display refresh request.
- Base:
    - A base request is a request that has the isochronous bit cleared.

Three VC sets are supported internally: low priority, display refresh, and high priority.

- Low priority: The low priority VC set consists of : posted, non-posted, and response.
- Display refresh: The display refresh VC set consists of: non-posted, and response.
- High priority: The high priority VC set consists of : posted, non-posted, and response.

Mapping between link and internal VC sets is described as follows:

- CPU downstream requests always use the link base channel.
- Peer-to-peer downstream uses the same downstream isoc type as the upstream request, except when modified by MSRC001_001F[EnConvertToNonIsoc].
- Upstream requests map to the three VC sets according to Table 3. A downstream response to an upstream request uses the same link channel as the request.

The following table defines how the F0x68[DispRefModeEn] and F0x1D0[HiPriModeEn] control how each of the 3 request types are routed to the 3 VC sets.

Table 3: Upstream display refresh and isochronous request routing

F0x68 [DispRef-ModeEn]	F0x1D0 [HiPri-ModeEn]	Definition
0	0	• All requests routed to the low priority channel. • F0x84[IsocEn] must be 0.
1	0	• Display refresh requests routed to the display refresh channel. • Base and isochronous requests routed to the low priority channel. • F0x84[IsocEn] must be 0.
0	1	• Isochronous and display refresh requests routed to the high priority channel. • Base requests routed to the low priority channel.
1	1	• Display refresh requests routed to the display refresh channel. • Isochronous requests routed to the high priority channel. • Base requests routed to the low priority channel.

2.6.3.2.1.1 Buffer Allocation and Programming Rules

After boot, buffer allocation and configuration may be changed by BIOS according to the following rules:

- Configuration changes should be made in the following order:
  - New values written to the following take effect after a warm reset: (no order required)
    - F0x90, F0x94, F0x1A4, F0x1D4, F3x6C, F3x74, F3x7C (all fields).
    - F0x68[DispRefModeEn]
    - F0x84[IsocEn]
    - F0x1D0[HiPriModeEn]
  - Warm reset.
  - New values written to the following take effect immediately:
    - F2x118 (all fields).
- If F0x84[IsocEn]=1 then F0x1D0[HiPriModeEn] must be 1.
- HTG upstream command buffer allocation rule:
  - F0x90[NpReqCmd + PReq + RspCmd] + F0x94[IsocNpReqCmd + IsocPReq + IsocRspCmd] <=36.
- HTG upstream data buffer allocation rule:
  - F0x90[RspData + NpReqData + PReq] + F0x94[IsocRspData + IsocNpReqData + IsocPReq] <= 20.
- HTG downstream ONION buffer allocation rule:
  - F0x1A4[DnHiRespBC + DnHiNpreqBC + DnHiPreqBC + DnLoRespBC + DnLoNpreqBC + DnLoPreqBC] + F0x1D4[DnDRRespBC + DnPoolBC] <= 34.
- NB upstream buffer allocation rules:
  - F3x6C[UpLoPreqDBC + UpLoNpreqDBC + UpHiPreqDBC + UpHiNpreqDBC] <= 8.
  - F3x74[UpHiRespCBC + UpLoRespCBC] <= 8.
  - F3x74[UpHiPreqCBC + UpHiNpreqCBC + UpLoPreqCBC + UpLoNpreqCBC + UpDRReqCBC] <= 16.
- NB in-flight buffer allocation rules:
  - F3x7C[FreePoolBC + DispRefBC + HiPriNPBC + HiPriPBC + LoPriNPBC + LoPriPBC + CpuBC] <= 16.
  - F3x7C[FreePoolBC] >= 1.
- Minimum allocation rules:
  - The following table lists, for each VC set, the buffers that must be sized >= 1.

Table 4: Minimum buffer allocation for each VC set

VC Set	Definition
Low Priority	• F0x90[PReq, NpReqCmd, NpReqData, RspCmd, RspData] • F3x6C[UpLoNpreqDBC, UpLoRespDBC, UpLoPreqDBC] • F3x74[UpLoRespCBC, UpLoNpreqCBC, UpLoPreqCBC] • F3x7C[LoPriPBC, LoPriNPBC]
Display Refresh	• F0x1D4[DnDRRespBC] • F3x74[UpDRReqCBC] • F3x7C[DispRefBC]
High Priority	• F0x94[IsocPReq, IsocNpReqCmd, IsocNpReqData, IsocRspCmd, IsocRspData] • F3x6C[UpHiNpreqDBC, UpHiRespDBC, UpHiPreqDBC] • F3x74[UpHiRespCBC, UpHiNpreqCBC, UpHiPreqCBC] • F3x7C[HiPriPBC, HiPriNPBC]

2.6.3.2.1.2 Buffer Allocation Recommendations

- The following settings are recommended:
  - Table 5: [F0x[94:90] Recommended Settings].

- Table 6: [F0x1A4 and F0x1D4 Recommended Settings].
- Table 7: [F3x6C and F3x74 Recommended Settings].
- Table 8: [F3x7C Recommended Settings].

The following abbreviations are defined for the following tables:

- NFCM is F0x84[IsocEn]=0.
- IFCM is F0x84[IsocEn]=1.
- DR is F0x68[DispRefModeEn]=1.
- P2PI is MSRC001_001F[EnConvertToNonIsoc]=0. (BIOS recommendation is 1).

Table 5: F0x[94:90] Recommended Settings

Register/Field		NFCM	IFCM, ~DR, ~P2PI	IFCM, DR, ~P2PI	IFCM, ~DR, P2PI	IFCM, DR, P2PI
F0x84[IsocEn]		0b	1b	1b	1b	1b
F0x68[DispRefModeEn]		x	0b	1b	0b	1b
MSRC001_001F[EnConvertToNonIsoc]		x	1b	1b	0b	0b
F0x90	RspData	2h	1h	1h	1h	1h
	NpReqData	1h	1h	1h	1h	1h
	RspCmd	2h	1h	1h	1h	1h
	PReq	11h	10h	Eh	Fh	Dh
	NpReqCmd	11h	11h	Dh	11h	Dh
F0x94	IsocRspData	0h	0h	0h	1h	1h
	IsocNpReqData	0h	1h	1h	1h	1h
	IsocRspCmd	0h	0h	0h	1h	1h
	IsocPReq	0h	1h	1h	1h	1h
	IsocNpReqCmd	0h	1h	7h	1h	7h

Table 6: F0x1A4 and F0x1D4 Recommended Settings

Register/Field	Enabled VC Sets					
	Lo	Lo, DR	Lo, Hi, ~P2PI	Lo, Hi, P2PI	Lo, DR, Hi, ~P2PI	Lo, DR, Hi, P2PI
F0x68[DispRefModeEn]	0b	1b	0b	0b	1b	
F0x1D0[HiPriModeEn]	0b		1b	1b	1b	
MSRC001_001F[EnConvertToNonIsoc]	0b		1b	0b	1b	0b

Table 6: F0x1A4 and F0x1D4 Recommended Settings

Register/Field		Enabled VC Sets					
		Lo	**Lo, DR**	**Lo, Hi, ~P2PI**	**Lo, Hi, P2PI**	**Lo, DR, Hi, ~P2PI**	**Lo, DR, Hi, P2PI**
F0x1A4	DnHiRespBC	0h		1h			
	DnHiNpreqBC	0h		0h	1h	0h	1h
	DnHiPreqBC	0h		0h	1h	0h	1h
	DnLoRespBC	Ah					
	DnLoNpreqBC	1h					
	DnLoPreqBC	4h	1h	4h		1h	
F0x1D4	DnDRRespBC	0h	6h	0h	0h	6h	6h
	DnPoolBC	13h	10h	12h	10h	0Fh	0Dh

Table 7: F3x6C and F3x74 Recommended Settings

Register/Field		Enabled VC Sets			
		Lo	**Lo, DR**	**Lo, Hi**	**Lo, DR, Hi**
F0x68[DispRefModeEn]		0b	1b	0b	1b
F0x1D0[HiPriModeEn]		0b	0b	1b	1b
F3x6C	UpHiRespDBC	8h		8h	
	UpHiNpreqDBC	0h		1h	
	UpHiPreqDBC	0h		1h	
	UpLoRespDBC	8h		8h	
	UpLoNpreqDBC	1h		1h	
	UpLoPreqDBC	7h		5h	
F3x74	UpDRReqCBC	0h	~(EMP | HT<=1400): 3h (EMP | HT<=1400): 4h	0h	~(EMP | HT<=1400): 3h (EMP | HT<=1400): 4h
	UpHiRespCBC	0h	0h	2h	2h
	UpHiNpreqCBC	0h	0h	1h	1h
	UpHiPreqCBC	0h	0h	1h	1h
	UpLoRespCBC	8h	8h	6h	6h
	UpLoNpreqCBC	9h	~(EMP | HT<=1400): 7h (EMP | HT<=1400): 6h	9h	~(EMP | HT<=1400): 6h (EMP | HT<=1400): 5h
	UpLoPreqCBC	7h	6h	5h	5h

Table 8: F3x7C Recommended Settings

Register/Field	Enabled VC Sets			
	Lo	**Lo, DR**	**Lo, Hi**	**Lo, DR, Hi**
F0x68[DispRefModeEn]	0	1	0	1
F0x1D0[HiPriModeEn]	0	0	1	1
FreePoolBC	Dh	5h	Bh	64B: 4h 32B: 3h
DispRefBC	0h	8h	0h	64B: 7h 32B: 8h
HiPriNPBC	0h		1h	
HiPriPBC	0h		1h	
LoPriNPBC	1h			
LoPriPBC	1h			
CpuBC	1h			

2.6.4 Memory Scrubber

DRAM memory is not scrubbed.

Systems that enable scrubbing should ensure that data-cache and L2 scrubbing 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][CpuDid]] 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.

2.6.5 Physical Address Space

The processor supports 40 address bits of coherent memory space (1 terabyte) 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[63:40].

2.6.6 System Address Map

System software must not map memory in the reserved link address regions. The link specification details the address map available to system hosts and devices. Downstream host accesses to reserved link address regions result in a page fault. Upstream system device accesses to reserved link address regions result in undefined operation.

2.7 Link

A *link* is a block of link signals, including 16 CAD signals, 2 CTL signals, and 2 CLK signals. A link operates per the *HyperTransport™ I/O Link Specification*, which will be referred to as the “link specification”. The electrical definition terminology for these modes is as follows:

- Gen1: refers to link rates of 0.4 to 1.6 GT/s in the revision 1 link specification or 2.0 GT/s in the revision 2 link specification.
- Gen3: refers to link rates of 2.4 to 5.2 GT/s in the revision 3 link specification. Note: 2.4 GT/s and 2.8 GT/s

are supported as specified in the revision 3 link specification only, not as specified in the revision 2 link specification.

2.7.1 Link Initialization

2.7.1.1 Link Type Detect

The link may be initialized in one of the following states during cold reset:
- The link must be ganged. Unganged is not supported.
- The link must be connected to another device through DC-coupled termination. AC-coupled termination is not supported.
- The link must be connected to the IO Hub through a chain or 0 or more devices. A disconnected link is not supported.

2.7.1.2 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)
- One 8-bit device connected to lower sublink and the upper sublink with inputs terminated.
- One 8-bit device connected to lower sublink and the upper sublink with inputs floating.

2.7.2 Termination and Compensation

The link is designed to operate in DC-termination mode as follows.

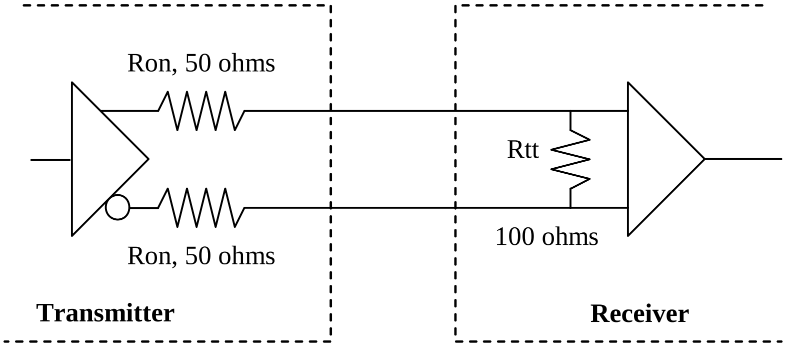

Figure 3: Link DC termination mode

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] F4x184_xE0[RonRawCal] and F4x184_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).

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 link. 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] F4x184_x[D5,C5]. Equalization is not used at Gen1 frequencies.

2.7.4 Link Retry

The link supports the error-retry mode described by the link specification, controlled by [The Link Retry Register] F0x130 and [The Link Global Retry Control Register] F0x150. Some notes and requirements about this mode:

- The processor does not support error-retry mode over the link operating at Gen1 frequencies.
- The IO link operating at Gen3 frequencies is required to have error-retry enabled.
- The retry history buffer is a shared structure with the downstream link buffer. It contains space for up to 25 packets (each packet may include command and data).

2.7.5 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), F0x84[LdtStopTriEn], and F0x170[RxLSSel] as follows:

Table 9: Link disconnect controls

Link Gen	LdtStopTriEn	RxLSSel	CLK	CAD, CTL	Reconnect delay
Gen1	0	X	L0[1]	L0[1]	Fast (about 1 us)[4]
Gen1	1	0xb	L0[1]	High imp[3]	Fast (about 1 us)[4]
Gen1	1	1xb	High imp[3]	High imp[3]	F3xD8[ReConDel][4]
Gen3	X	00b	L0[1]	EI[2]	F0x16C[T0Time], unless there is a frequency change, or if the active lanes exceed the idle timeout (as indicated by F0x16C[ForceFullT0]), then the reconnect delay is specified by F0x16C[FullT0Time].
Gen3	X	01b	L0[1]	L0[1]	
Gen3	X	10b	EI[2]	EI[2]	

Notes:
1. L0 represents the active, driven state.
2. Electrical idle.
3. High impedance.
4. F0x84[ExtCTL]=1 adds 50us after CTL asserts.

2.7.6 LDTSTOP Requirements

The LDTSTOP requirements are:

- The processor requires additional minimum LDTSTOP_L assertion time for the following system configurations; otherwise the minimum LDTSTOP_L assertion time is as specified by the link specification (1 us).
  - If a link in the system is operating at a Gen3 frequency and the LDTSTOP_L assertion is associated with a link frequency change, the LDTSTOP_L assertion time required by the processor is 1 us plus the link PLL lock time (F3xD4[LnkPllLock]).
- For all cases of LDTSTOP_L triggered by the STOPGRANT message (including link width/frequency changes, S1-based power management) LDTSTOP_L must not deassert less than 10 us after the processor

broadcasts the STOPGRANT message.

- Minimum LDTSTOP_L deassertion time:
  - The processor requires a minimum LDTSTOP_L deassertion time of 3 us if:
    - F2x[1,0]A0[DllVRegPwrDwn]=00b.
  - The processor requires a minimum LDTSTOP_L deassertion time of 5 us if:
    - F2x[1,0]A0[DllVRegPwrDwn]!=00b.
- Note that narrow links, slow links, and use of F0x84[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 link specification, titled “Disconnecting and Reconnecting HyperTransport™ Links“.

2.7.7 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 that target system memory 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, unless F0x8C[UnitIdReOrderDis]=1.
- The requests have the same, non-zero SeqID value.
- The requests have the same PassPW bit value.
- The requests have the same RespPassPW bit value.
- The requests have the same Coherent (snoop) 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].

2.7.8 Link Testing, BIST, and ILM

The processor includes a link-defined BIST engine for the link. The control registers are found starting at [The Link BIST Control Register] F4x184_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] F0x170[ILMEn].

2.7.9 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.
- 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 the IO link is supported (i.e., setting LinkTrain[DisCmdThrt] on the other side of the link) and command packet insertion from the IO link 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[DisNcHtC-

mdThrottle].

- The processor logically supports link-defined mode combinations as follows (however, electrical requirements may limit some options):

Table 10: Supported link operational modes

Frequency	200-1000MHz	1200-2600MHz	
Coupling/ Link Type	DC	DC non-coherent operational	DC test/debug
Termination	RXDIFF	RXDIFF	RXDIFF
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 from the link. 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.

2.7.10 LDTREQ_L

The processor asserts LDTREQ_L whenever a transaction is queued upstream or downstream. See link specifications.

2.8 DRAM Controller (DCT)

The DCTs supports the following types of DDR2 DIMMs:

- unbuffered DIMMs
- SO-DIMMs
- uDIMMs

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.

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).

When the DCTs are in ganged mode, as specified by [The DRAM Controller Select Low Register] F2x110[DctGangEn], then 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]).

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

- All DIMMs connected to a processor are required to operate at the same MEMCLK frequency, regardless of which channel they are connected to. Both DCTs must be programmed to the same frequency.
- The DCTs do not support registered DIMMs.
- x4 (by 4) DIMMs are not supported.
- Quad rank DIMMs are not supported.
- A maximum of two, 2-Rank SODIMMs, can be supported on DCT0 at one time. DCT1 is unpopulated in this case.
- A single SODIMM per DCT can be supported.
- A single SODIMM and one single rank soldered down DRAM can be supported per DCT.

Furthermore, the processor logically supports the memory configurations per channel as described by Table 11.

Table 11: Supported Memory Configurations

DIMM 0		DIMM 1	
CS0	**CS1**	**CS2**	**CS3[2]**
SR or DR		-	-
-	-	SR or DR[2]	
SR or DR		SR or DR[2]	
SR or DR		SD	-
-	-	SD	-
Notes: 1. SR = Single rank. DR = Dual rank. SD = soldered down = Single rank memory device is soldered onto the motherboard. 2. CS3 on DCT1 is not supported.			

The following figures show the supported memory configurations and DIMM signal connections from the processor. The DIMMs on each channel are numbered from 0 to n where DIMM0 is the DIMM closest to the processor on that channel and DIMM1 is the DIMM farthest from the processor on that channel. DIMMs must be populated from the farthest slot to the closest slot to the processor on a per channel basis beginning with channel A. Single soldered down DRAM devices should always use CS2. Figure 4 shows a single channel with two dual rank DIMMs. In this case, channel B must be unpopulated and F2x[1,0]94[CkeOdtMap] = 0. Two dual rank DIMMs on channel B is not supported.

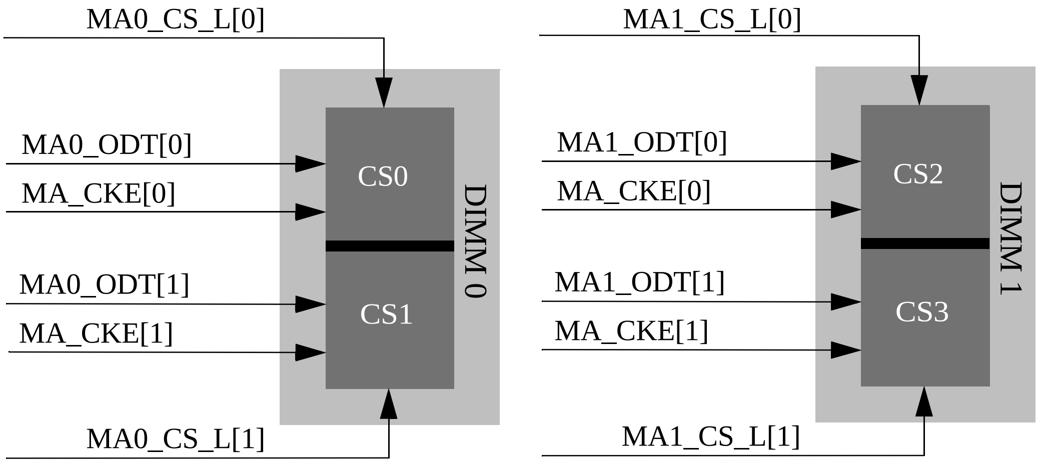

Figure 4: Single channel A with 2 dual rank DIMMs

Figure 5 shows a single channel, A or B, with one dual rank DIMM and 1 single rank soldered down. In this case, both channels may have this configuration. See Table 11 for other supported variations. Note that in this case, if F2x[1,0]94[PowerDownEn] = 1 and F2x[1,0]94[PowerDownMode] = 1, any access to CS1 requires termination to be supplied from the soldered down DRAM and thus the channel behaves as if it was in channel CKE control mode. See F2x[1,0]94[PowerDownMode].

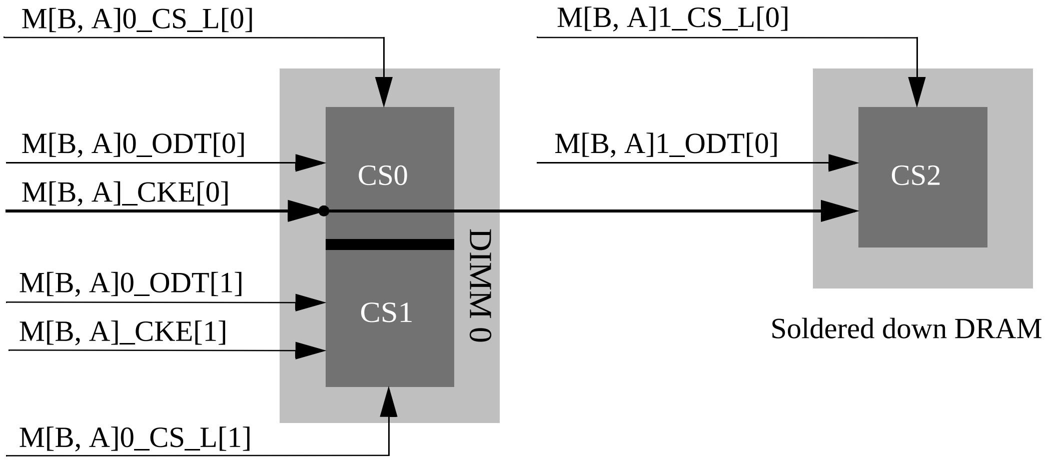

Figure 5: Channel A or B with 1 dual rank DIMM and 1 single rank down

The next two figures describe the cases where power savings is enhanced by remapping the ODT signals in conjunction with F2x[1,0]94[CkeOdtMap]. Figure 6 shows a single channel with two dual rank DIMMs and the ODT signals remapped. In this case, channel B must be unpopulated and F2x[1,0]94[CkeOdtMap] = 1.

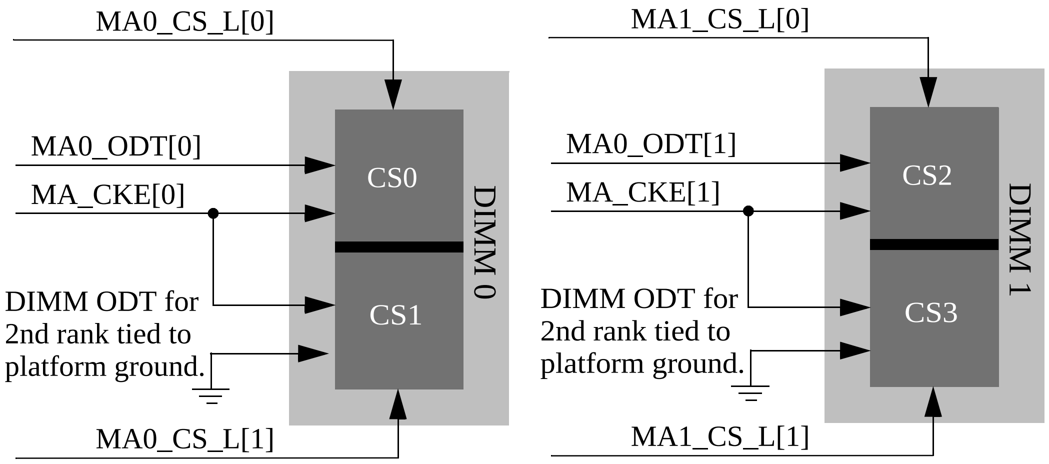

Figure 6: Single channel A with 2 dual rank DIMMs (CkeOdtMap = 1)

Figure 7 shows a dual channel remapped ODT configuration with two dual rank DIMMs and one single rank soldered down DRAM device. In this case, F2x[1,0]94[CkeOdtMap] = 1. Note that for all multi-DIMM systems where F2x[1,0]94[PowerDownEn] = 1 and F2x[1,0]94[CkeOdtMap] = 1, the channel degrades to channel CKE control mode. See F2x[1,0]94[PowerDownMode].

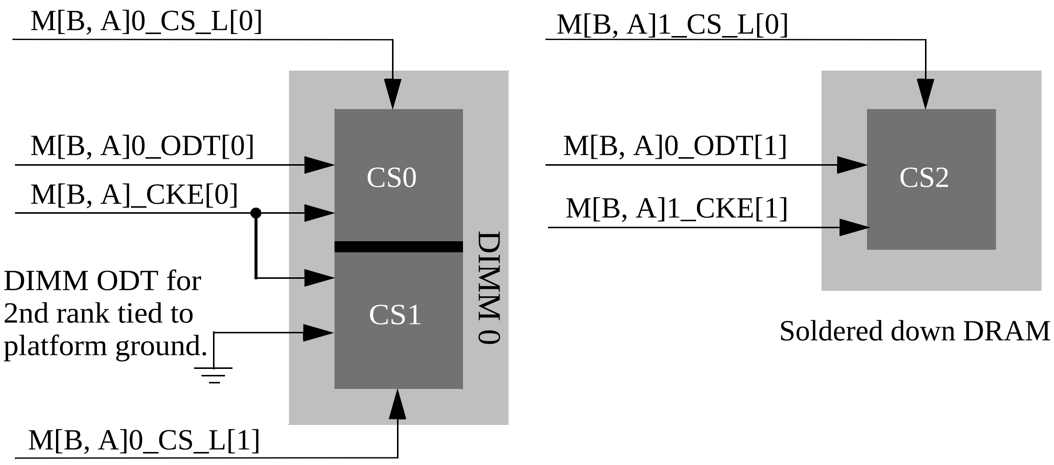

Figure 7: Channel A or B with 1 dual rank DIMM and 1 single rank down (CkeOdtMap = 1)

2.8.1 DCT Configuration Registers

DCT configuration registers range from F2x[1,0][4C:40] through F2xA4 and F2x110 through F2x11C. F2x0xx registers are associated with DCT0 and F2x1XX registers are associated with DCT1.

When the DCTs are ganged (F2x110[DctGangEn]=1), as specified by F2x110[DctGangEn], then the DCT configuration registers F2x[1,0][A0,94:00] behave as follows:

- The F2x[1,0][A0,94:00] registers can be read and written.
- The F2x0[A0,94:00] registers are applied to both DCT0 and DCT1. See exceptions below.
- The F2x1[A0,94:00] registers are ignored and do not need to be initialized. See exceptions below.

For the following list of register fields, DCT0 and DCT1 must be initialized to the same value, independent of the setting of F2x110[DctGangEn] or if one of the DCT channels is disabled:

- F2x[1,0]90[IdleCycInit, DynPageCloseEn]
- F2x[1,0]94[DcqBypassMax]
- F2x[1,0]A0[PhyClkDivInSR]

Note that the DCT phy registers, F2x[1,0]98, F2x[1,0]9C, and all the associated indexed registers (F2x[1,0]9C_xXX); 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]), only 64-byte bursts are supported.

2.8.4 Routing DRAM Requests

DRAM requests are mapped to the DCT based on the routing configuration specified in section 2.6.3.1.1 [DRAM and MMIO Memory Space]. They are mapped to chip selects through [The DRAM CS Base Address Registers] F2x[1,0][4C:40], and [The DRAM CS Mask Registers] F2x[1,0][64:60].

The following algorithm is designed to be used to determine 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, ChannelSelect, and CS are output variables. If CSFound is equal to 1, then ChannelSelect, and CS outputs are equal to the DRAM controller (zero or one), and the chip select that corresponds to the input address.

```
(int,int,int) TranslateSysAddrToCS((uint64)SystemAddr){

int CSFound, CS, F2Offset, F2MaskOffset;
int HiRangeSelected;
int ChannelSelect;
uint32 HoleOffset, HoleEn;
uint32 CSBase, CSLimit, CSMask, CSEn;
uint32 InputAddr, Temp;
uint32 DctSelBaseAddr, DctSelIntLvAddr, DctGangEn, DctSelIntLvEn;
uint32 DctSelHiRngEn,DctSelHi;
uint64 DctSelBaseOffsetLong, ChannelOffsetLong,ChannelAddrLong;

CSFound = 0;
HoleEn = Get_PCI(bus0, dev24, func1, 0xF0);
HoleOffset = (HoleEn & 0x0000FF80);
HoleEn = (HoleEn &0x00000003);
Temp = Get_PCI(bus0, dev24, func2, 0x110);
DctSelHiRngEn = Temp & 1;
DctSelHi = Temp>>1 & 1;
```

```
DctSelIntLvEn = Temp & 4;
DctGangEn = Temp & 0x10;
DctSelIntLvAddr = (Temp>>5) & 3;
DctSelBaseAddr = Temp & 0xFFFFF800;
DctSelBaseOffsetLong = Get_PCI(bus0, dev24, 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 = SystemAddr>>16&0x1F;
   switch(Temp){
      case 1:
      case 2:
      case 4:
      case 8:
      case 16:Temp=1;break;
      default :Temp=0;break;
   }
   ChannelSelect = (SystemAddr>>6)^Temp;
}
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 & 0xFF_F8000000;
}

//Remove hoisting offset and normalize to DRAM bus addresses
ChannelAddrLong = SystemAddr & 0x0000FFFF_FF800000 -
ChannelOffsetLong & 0xFFFFFFFF_FF800000;

//Remove Channel interleave and hash
if(DctSelIntLvEn && DctSelHiRngEn==0 && DctGangEn==0)
{
   if(DctSelIntLvAddr != 1)
      ChannelAddrLong = (ChannelAddrLong>>1) & 0xFFFFFFFF_FFFFFFC0;
   else
   {
      Temp = ChannelAddrLong & 0xFC0;
      ChannelAddrLong = ((ChannelAddrLong & 0xFFFFFFFF_FFFFC000) >> 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, func2, F2Offset);
   CSEn = CSBase & 0x00000001;
   CSBase = CSBase & 0x1FF83FE0;
   CSMask = Get_PCI(bus0, dev24, func2, F2MaskOffset);
   CSMask = (CSMask | 0x0007C01F) & 0x1FFFFFFF;
   if(CSEn && ((InputAddr & ~CSMask) == (CSBase & ~CSMask)))
      CSFound = 1;
}
return(CSFound,ChannelSelect,CS);
```

2.8.5 DRAM Data Burst Mapping

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

- 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:
  - Channel A: QW0, QW1, QW4, QW5.
  - Channel B: QW2, QW3, QW6, QW7.

2.8.6 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. DRAM controller and device initialization.
   a. Program SPD timings. See section 2.8.6.1.
   b. Program Non-SPD timings. See section 2.8.6.2 and all sub-sections.
   c. DRAM device initialization. See section 2.8.6.3.
2. DRAM Data Training. See section 2.8.6.4 and all sub-sections:
   d. Receiver Enable Training. See section 2.8.6.4.1.
   e. DQS Position Training.See section 2.8.6.4.2.
   f. MaxAsyncLat Training. See section 2.8.6.4.3.1.
3. Program Non-SPD timings to optimal values. See section 2.8.6.2 and all sub-sections.
4. The memory subsystem is ready for use.

After reset associated with suspend-to-RAM, the BIOS performs the steps listed in section [The ACPI Suspend to RAM State (S3)] 2.4.4.

2.8.6.1 SPD ROM-Based Configuration

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

BIOS reads the values encoded in the SPD ROM through the IO Hub, which obtains the information through a secondary device connected to the IO Hub through the SMBus. This secondary device communicates with the DIMM by means of the I^2C bus. Typically, system BIOS acquires DIMM configuration information, such as the amount of memory on each DIMM, from the SPD ROM on each DIMM and uses this information to program the DRAM controller registers.

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

- 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 2.8.6.3 [DRAM Device Initialization] for more information on configuring the bus frequency.

2.8.6.2 Non-SPD ROM-Based Configuration

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

- Training delay values. See section 2.8.6.4 [DRAM Training].
- Read and write latency differences.
- The phy's idle clock requirements on the data bus.

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.

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

This timing parameter, F2x[1,0]8C[Twrrd], accounts for termination timing when a write is followed by a read to a different DIMM. Prior to DRAM training, BIOS should program F2x[1,0]8C[Twrrd]=11b; otherwise, BIOS should program this value optimally after DDR training is complete based on the following table:

Table 12: Twrrd settings

CD[1] in MEMCLKs	Recommended Twrrd value	Virtual CAS (write) to CAS (read) separation in MEMCLKs	Idle cycle(s) on the data bus in MEMCLKs
0	01b	2	2
0.5	01b	2	2.5
1.0	00b	1	2
1.5	00b	1	2.5
2.0	00b	1	3
2.5	00b	1	3.5
3.0	00b	1	4
3.5	00b	1	4.5

[1] The largest F2x[1,0]9C_x[02:01][DqsRcvEnGrossDelay] delay for any DIMM on the channel is equal to the Critical Delay (CD) for Twrrd.

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

This timing parameter, F2x[1,0]8C[TrwtTO], ensures read-to-write data-bus turnaround. Prior to DRAM training, BIOS should program F2x[1,0]8C[TrwtTO]=0111b ; otherwise, BIOS should program this value optimally after DDR training is complete as follows:

- BIOS should program the optimal TrwtTO based on the following table:

Table 13: TrwtTO (Read-to-Write Turnaround for Data, DQS Contention)

CD[1] in MEMCLKs	Recommended TrwtTO value	Virtual CAS (read) to CAS (write) separation in MEMCLKs	Idle cycle(s) on the data bus in MEMCLKs
0.5	0010b	4	1
1.0	0011b	5	1.5
1.5	0011b	5	1.0
2.0	0100b	6	1.5
2.5	0100b	6	1.0
3.0	0101b	7	1.5
3.5	0101b	7	1.0
4.0	0110b	8	1.5

[1] The BIOS picks the largest F2x[1,0]9C_x[02:01][DqsRcvEnGrossDelay] delay for any DIMM on the channel and rounds the value up to the nearest one-half MEMCLK by observing the DIMM's associated DqsRcvEnFineDelay value. After rounding, this value is equal to the Critical Delay (CD) for TrwtTO.

2.8.6.2.3 FourActWindow (Four Bank Activate Window or tFAW)

No more than 4 banks may be activated in a rolling tFAW window, as configured by F2x[1,0]94[FourActWindow]. To program this field, BIOS must convert the tFAW parameter into MEMCLK cycles by dividing the highest tFAW parameter (in ns) found in all the DIMMs connected to the channel by the period of MEMCLK (in ns) and rounding up to the next integer. As an example of the rolling window, if (tFAW/tCK) rounds up to 10 clocks, and an activate command is issued in clock N, no more than three further activate commands may be issued in clock N+1 through N+9. Table 14 shows the F2x[1,0]94[tFAW] clock values used for various frequencies and page sizes.

Table 14: Four Bank Activate Window Values

Page Size	400 MHz	333 MHz	266 MHz
1K	14 MEMCLKs	13 MEMCLKs	10 MEMCLKs
2K	18 MEMCLKs	17 MEMCLKs	14 MEMCLKs

2.8.6.2.4 DRAM ODT Control

This section describes the ODT configurations and settings for the processor and DIMMs. Table 15 documents the ODT termination values on a per channel basis for different DDR DIMM speeds.

Table 15: Processor and DIMM ODT Settings

DDR Type-Rate	Number of DIMMs	Processor ODT	DIMM ODT
DDR2-533, DDR2-667, DDR2-800	1	150 ohms	150 ohms
DDR2-533, DDR2-667	2	150 ohms	75 ohms
DDR2-800	2	150 ohms	50 ohms

The following describes the behavior of the ODT signals for the DIMMs (1 or 2 DIMMs on a channel):

- For writes
  - ODT is active for the first rank of the non-targeted DIMM.
  - All other DIMM rank ODTs are off.
  - If there is only one DIMM on the channel, then the first rank of that DIMM provides ODT.
  - The processor ODT is off.
- For reads:
  - ODT is active for the first rank of the targeted DIMM.
  - All other DIMM ODTs are off.
  - If there is only one DIMM on the channel, then DIMM ODT is off.
  - The processor ODT is on.

2.8.6.2.5 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 for 1 single or dual rank DIMM configurations. The settings for the other supported DIMM configurations, 2 DIMMs or 1 DIMM and 1 rank soldered down, must be determined by the customer based on system timing analysis.

Table 16 documents the address timing and output driver settings on a per channel basis for different DDR DIMM types. Populations that are not shown in these tables are not supported.

Table 16: SO-DIMM Address Timings and Drive Strengths for S1g2 Package

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

1. SR = Single Rank DIMM
   DR = Dual Rank DIMM
   See 2.8 [DRAM Controller (DCT)] for further processor restrictions on DRAM population for both channels.
2. See 2.8.6.2.4 [DRAM ODT Control] for ProcOdt settings.

2.8.6.3 DRAM Device Initialization

After the DCT registers are programmed (see sections 2.8.6.1 and 2.8.6.2), BIOS initializes the DRAM using either a hardware or BIOS controlled sequence. See sections 2.8.6.3.1 [Software DDR2 Device Initialization] for more information on BIOS controlled initialization. To initiate the hardware sequence, BIOS sets F2x[1,0]90[InitDram]=1. 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).

When both DCTs are enabled in unganged mode, BIOS must initialize each DCT in order after the registers for both channels are programmed. This is accomplished as follows:
- Configure all DCT registers with F2x[1,0]94[MemClkFreq]=0.
- Program F2x[1,0]94[MemClkFreq]=1.
- For HW initialization: Program F2x090[InitDram]=1 then F2x190[InitDram]=1 back-to-back and ensuring that no interrupt or exception can be taken between the two config writes.
  For SW initialization: Program F2x07C[EnDramInit]=1 then F2x17C[EnDramInit]=1 back-to-back and ensuring that no interrupt or exception can be taken between the two config writes when using SW DRAM initialization.

A similar requirement exists for changing the frequency of each DCT:
- Program F2x[1,0]94[MemClkFreqVal]=0.
- Program F2x[1,0]94[MemClkFreq] to the proper operating frequency for the DCTs.
- Program F2x94[MemClkFreqVal]=1.

Part of the initialization sequence includes writing mode register set (MRS) values to the 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[BurstLength32] and F2x110[DctGangEn].
- 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 disable: fixed at 0. (enabled).
- EMRS(1)[11]: RDQS enable: fixed at 0. (disabled).
- EMRS(1)[12]: Qoff: output buffer enabled.
- EMRS(2)[7]: SRF: high temperature self refresh rate enable, based on F2x[1,0]90[SelfRefRateEn].

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.

2.8.6.3.1 Software DDR2 Device Initialization

The following BIOS controlled software initialization procedure applies to each DRAM controller and will properly initialize 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.
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.

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 = *target chip select*; 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 the [The DRAM Initialization Register] F2x[1,0]7C register 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].
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 the [The DRAM Initialization Register] F2x[1,0]7C register 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 the [The DRAM Initialization Register] F2x[1,0]7C register 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.

2.8.6.4 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 2.8.6.3 [DRAM Device Initialization]). DRAM training is entirely BIOS controlled and 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 reads and writes. Part of the training process involves a latency component called F2x[1,0]94[MaxAsyncLat]. This is described in section 2.8.6.4.3 [Maximum Asynchronous Latency (MaxAsyncLat)].

If the DCTs are to be operated in ganged mode (see section 2.8 [DRAM Controller (DCT)]) 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]94[MaxAsyncLat] 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.

Before DRAM training BIOS should configure the processor to the P0 state. See section 2.4.2 [P-states]. When DRAM training is complete BIOS can return the processor to the desired operating P-state.

2.8.6.4.1 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. DQS receiver enable training is done per channel, per DIMM. The second phase must be done after DQS position training is complete. See 2.8.6.4.2 [DQS Position Training] for details on how to train the DRAM for proper DQS position.

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

For each channel:
1. Program [The DRAM Write Data Timing [High:Low] Registers] F2x[1,0]9C_x[02:01] to 00h for all bytes.
2. Program [The DRAM Read DQS Timing Control [High:Low] Registers] F2x[1,0]9C_x[06:05] to 2Fh for all bytes.
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:
   Program the delay timing fields for each F2x[1,0]9C_x[2B:10] register with 0:

   a. Program F2x[1,0]94[MaxAsyncLat] with the current greatest value of F2x[1,0]9C_x[2B:10] plus 3 ns.
   b. Read the first test address for the current rank and compare the first data beat of the data read with 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 setting for the current rank to the corresponding F2x[1,0]9C_x[2B:10] register.
   d. Read the second test address for the rank and compare the first data beat of the data read with the value written in step 6 above.
   e. Save the DQS receiver enable setting that passes for both read patterns. Continue to step 8 below if all the data is read correctly; otherwise, continue to step f.
   f. Increment the current total delay DQS receiver enable setting by one in F2x[1,0]9C_x[2B:10] and repeat steps a through e.

8. 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 6/64 MEMCLKs of additional delay.
   - Save the first DQS receiver enable delay settings that pass for all ranks. This is used in phase two below.
9. Program F2x[1,0]94[MaxAsyncLat] using the largest F2x[1,0]9C_x[2B:10] register setting plus 3 ns.
10. 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 2.8.6.4.2 [DQS Position Training]. 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 the total delay setting for 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, plus 6/64 MEMCLKs of additional delay.
3. Write a cache line to the first and second test addresses 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
4. For each channel:
   - For each rank:
     a. Program the F2x[1,0]94[MaxAsyncLat] field with the current DQS receiver enable total delay setting plus 3 ns.
     b. Read the first test address for the rank and compare the data read with the written value from step 3 above.
     c. Reset the read pointer in the DRAM controller FIFO by writing the current corresponding DQS receiver enable delay settings in each corresponding F2x[1,0]9C_x[2B:10] register.
     d. Read the second test address for the current rank and compare the data read with the expected value from step 3 above.
     e. Reset the read pointer in the DRAM controller FIFO by writing the current corresponding DQS receiver enable delay settings in each corresponding F2x[1,0]9C_x[2B:10] register.
     f. Save the delay setting that passes for read patterns of both test addresses. Continue to step g if all

    the data is read correctly; otherwise, continue to step 5 below.
    g. Increment the current delay setting by one in F2x[1,0]9C_x[2B:10] and repeat steps a through f.

5. For each DIMM (chip select pair):
   For each delay setting in each F2x[1,0]9C_x[2B:10] register saved in phase 1, add each corresponding delay setting in each F2x[1,0]9C_x[2B:10] register saved in phase 2. Divide this value by two. This centers the DQS receiver enable in the passing window for each rank.
6. For each channel: Average the values calculated in step 5 for each enabled rank. This centers the DQS receiver enable in a passing window for all enabled ranks on the channel and is the final delay setting to be used for each F2x[1,0]9C_x[2B:10] register. This value is then programmed for each rank on the channel regardless of whether it is enabled.
7. Program the F2x[1,0]94[MaxAsyncLat] field using the largest total delay setting from F2x[1,0]9C_x[2B:10] in step 5 above plus 3 ns.

2.8.6.4.2 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[02:01]) and the DRAM Read DQS Timing Control registers (see F2x[1,0]9C_x[06:05]) registers for DQS position training.

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 cache lines 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 F2x[1,0]9C_x[02:01] setting of the current byte:
         - Write the current write DQS delay value to F2x[1,0]9C_x[02:01] for the current byte lane.
         - Write the DRAM training pattern to the first test address for the current rank.

           DRAM Read DQS Timing Control Loop:
           - For each read delay setting for F2x[1,0]9C_x[06:05]:
             a. Write the current F2x[1,0]9C_x[06:05] 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 all three times, record the read position for the current byte lane as a pass and exit the **DRAM Read DQS Timing Control Loop**.
             d. Increment the F2x[1,0]9C_x[02:01] setting for the current byte lane and continue in the **DRAM Read DQS Timing Control Loop**.
         - If the read DQ to DQS delay setting for the current byte lane contains three or more consecutive passing values, exit the **DRAM Write Data Timing Loop**.
         - Increment the F2x[1,0]9C_x[06:05] byte for the current byte lane and continue in the **DRAM Write Data Timing Loop**.
       - Write the F2x[1,0]9C_x[06:05] setting for the current byte with a value that represents the center position of the passing region.
       - Write 0 to F2x[1,0]9C_x[02:01] for the current byte lane.
       - For each F2x[1,0]9C_x[02:01] setting:
         a. Write the current F2x[1,0]9C_x[06:05] 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 F2x[1,0]9C_x[02:01] setting for the current byte lane as a pass.
   f. Increment the F2x[1,0]9C_x[02:01] setting 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 F2x[1,0]9C_x[06:05] for the current byte lane with the centered delay setting of the mutually passing region for reads.
- Write F2x[1,0]9C_x[02:01] for the current byte with the centered delay position of the mutual passing region for writes.

2.8.6.4.3 Maximum Asynchronous Latency (MaxAsyncLat)

The maximum asynchronous latency, F2x[1,0]94[MaxAsyncLat], is the maximum round-trip latency in the system from the processor to the DRAM devices and back. It is recommended that BIOS train for the optimal value for F2x[1,0]94[MaxAsyncLat]; see 2.8.6.4.3.1. The initial value of F2x[1,0]94[MaxAsyncLat] is the sum of the following two components:

1. The worst case delay setting for [The DRAM DQS Receiver Enable Timing Control Registers] F2x[1,0]9C_x[2B:10] in ns.
2. A constant of 3ns associated with the delay internal to the processor.

If the sum of these two components is less than 4ns, F2x[1,0]94[MaxAsyncLat] should be programmed to 4ns.

2.8.6.4.3.1 MaxAsyncLat Training

This section describes an algorithm that can be used to optimize F2x[1,0]94[MaxAsyncLat] value used after DRAM training:

The following three cache line patterns are used to train the F2x[1,0]94[MaxAsyncLat] 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

- If the channels are unganged:
  - The following MaxAsyncLat training algorithm described below is done separately for each channel.
  - The starting F2x[1,0]94[MaxAsyncLat] delay value is the worst case [The DRAM DQS Receiver Enable Timing Control Registers] F2x[1,0]9C_x[2B:10] setting found during training on either channel. Convert this setting to a nanosecond value.
- If the channels are ganged:
  - The following MaxAsyncLat training algorithm described below is done once for the ganged channels.
  - The starting F2x[1,0]94[MaxAsyncLat] delay value is the worst case [The DRAM DQS Receiver Enable Timing Control Registers] F2x[1,0]9C_x[2B:10] setting found during training on the channel being MaxAsyncLat trained. Convert this setting to a nanosecond value.
  - F2x094[MaxAsyncLat] applies to both channels. F2x194[MaxAsyncLat] is ignored.
- BIOS then selects an address associated with the DIMM with the worst case F2x[1,0]9C_x[2B:10] register setting that was found. Using the patterns given above, write 3 cache lines to the target address on the DIMM.
- According to the following steps, BIOS sets the initial value of MaxAsyncLat and increments MaxAsyncLat until the 3 cache lines are read successfully.
  1. Read the three cache lines from the selected addresses on the targeted DIMM.
  2. Compare all three cache lines of data.
     - If the compare matches then add the integer that equals “MEMCLK cycle in ns rounded up to the nearest integer” to MaxAsyncLat and go to step 3. below.
     - If the compare does not match, increment MaxAsyncLat and go to step 1. above.
  3. Save the current MaxAsyncLat value.

2.8.7 Memory Interleaving Modes

The processor supports two 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][4C:40]. CS 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.
- Channel: interleaving between the two 64-bit channels of a processor. This is controlled through [The DRAM Controller Select Low Register] F2x110[DctSelIntLvEn]. Channel interleaving is defined as the spreading contiguous physical address space over two channels, as opposed to each channel owning a single contiguous address space. This is accomplished by using lower-order address bits to select between channels.

Any combination of these interleaving modes may be enabled concurrently.

2.8.7.1 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][4C:40] and [The DRAM CS Mask Registers] F2x[1,0][64:60] in memory interleaving mode is as follows:

1. Program all DRAM CS Base Address and DRAM CS Mask registers using contiguous normalized address

mapping.

2. For each enabled chip select, swap the corresponding F2x[1,0][4C:40][BaseAddr[36:27]] bits with F2x[1,0][4C:40][BaseAddr[21:13]] bits, as defined by Table 17 and Table 18.
3. For each enabled chip select, swap the corresponding F2x[1,0][64:60][AddrMask[36:27]] bits with F2x[1,0][64:60][AddrMask[21:13]] bits, as defined by Table 17 and Table 18.

Table 17: Swapped normalized address lines for interleaving for a 64-bit interface

Chip Select Mode	**Chip Select Size**	**Swapped Base Address and Address Mask bits**	
		4 way CS interleaving	**2 way CS interleaving**
0000b	128-MB	[28:27] and [15:14]	[27] and [14]
0001b	256-MB	[29:28] and [16:15]	[28] and [15]
0010b	512-MB	[30:29] and [16:15]	[29] and [15]
0011b	512-MB	[30:29] and [17:16]	[29] and [16]
0100b	512-MB	[30:29] and [17:16]	[29] and [16]
0101b	1-GB	[31:30] and [17:16]	[30] and [16]
0110b	1-GB	[31:30] and [17:16]	[30] and [16]
0111b	2-GB	[32:31] and [17:16]	[31] and [16]
1000b	2-GB	[32:31] and [18:17]	[31] and [17]
1001b	4-GB	[33:32] and [18:17]	[32] and [17]
1010b	4-GB	[33:32] and [17:16]	[32] and [16]
1011b	8-GB	[34:33] and [18:17]	[33] and [17]

Table 18: 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**	
		4 way CS interleaving	**2 way CS interleaving**
0000b	256-MB	[29:28] and [16:15]	[28] and [15]
0001b	512-MB	[30:29] and [17:16]	[29] and [16]
0010b	1-GB	[31:30] and [17:16]	[30] and [16]
0011b	1-GB	[31:30] and [18:17]	[30] and [17]
0100b	1-GB	[31:30] and [18:17]	[30] and [17]
0101b	2-GB	[32:31] and [18:17]	[31] and [17]
0110b	2-GB	[32:31] and [18:17]	[31] and [17]
0111b	4-GB	[33:32] and [18:17]	[32] and [17]
1000b	4-GB	[33:32] and [19:18]	[32] and [18]
1001b	8-GB	[34:33] and [19:18]	[33] and [18]
1010b	8-GB	[34:33] and [18:17]	[33] and [17]
1011b	16-GB	[35:34] and [19:18]	[34] and [18]

The following is an example of interleaving a 64-bit interface to the DRAM. The DRAM memory consists of two 2-sided 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 17, 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][4C:40][BaseAddr[21:20]. Base address bits [16:15] are defined with F2x[1,0][4C: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][64:60][AddrMask[21:20]. Address mask bits [16:15] are defined with F2x[1,0][64:60][AddrMask[8:7].
   - F2x[1,0]60 = 0038_3E70h
   - F2x[1,0]64 = 0038_3E70h

2.8.8 Memory Hoisting

Memory hoisting is defined as reclaiming (relocating) the unusable 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 region of DRAM that is hoisted is defined to be from F1xF0[DramHoleBase] to the 4G level. The beginning of the upper memory space, as specified by F2x110[DctSelHi, DctSelBaseAddr], is allowed to occur at any relationship with respect to the MMIO hole. Hoisting raises the hoisting disabled memory configuration from DramHoleBase and above up to the 4G level. This applies to both non-interleaved and interleaved configurations.

The memory hoisting offset fields, F1xF0[DramHoleOffset] and F2x114[DctSelBaseOffset], are programmed based on the following parameters:

- F1xF0[DramHoleBase] specifies the base address of the MMIO hole below the 4G level.
- F2x110[DctSelBaseAddr] specifies the base address of the upper memory space owned by the DCT specified by F2x110[DctSelHi].
- F2x110[DctSelIntLvEn] specifies if interleaving between the two DCTs is enabled (channel interleave mode).
- F2x[1,0][4C:40][CSEnable] specifies if one or both DCTs are enabled. 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]).

2.8.8.1 DctSelBaseAddr Programming

F2x110[DctSelBaseAddr], for both interleaved and non-interleaved, is adjusted for hoisting based on

F1xF0[DramHoleBase] and the hoisting disabled value of DctSelBaseAddr, called (non-hoisted-DctSelBaseAddr). For the remainder of this section DctSelBaseAddr will refer to the hoisting enabled value of DctSelBaseAddr.

DctSelBaseAddr is programmed to one of the following equations based on the scenario:

- **Case 1**: for any of the following conditions:
  - non-hoisted-DctSelBaseAddr >= DramHoleBase. (Non-hoisted-DctSelBaseAddr[39:27] >= {00h,DramHoleBase[31:27]})

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

  then:
  - DctSelBaseAddr is unchanged.

2.8.8.2 DramHoleOffset Programming

F1xF0[DramHoleOffset] is programmed to one of the following equations based on the scenario:

- **Case 1**: for any of the following conditions:
  - 1 DCT is enabled
  - 2 DCTs are enabled in non-interleaved mode and (non-hoisted-DctSelBaseAddr > DramHoleBase)
  - 2 DCTs are enabled in interleaved mode and the same size (F2x110[DctSelHiRngEn]=0)
  - 2 DCTs are enabled in interleaved mode and (non-hoisted-DctSelBaseAddr > DramHoleBase)

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

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

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

  then:
  - DramHoleOffset[31:23] is not used.

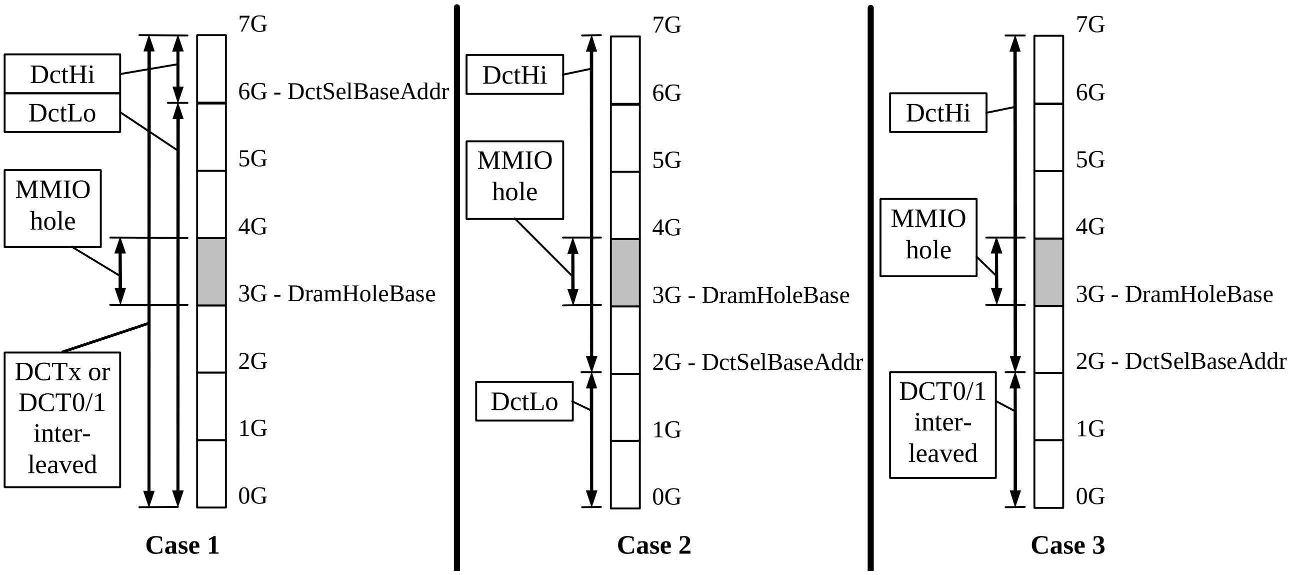

Figure 8: Example cases for programming DramHoleOffset

2.8.8.3 DctSelBaseOffset Programming

F2x114[DctSelBaseOffset] is programmed to one of the following equations based on the scenario:

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

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

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

  then:
  - DramHoleBase[26:24] must be 000b.
  - DctSelBaseOffset[39:26] = {00h,DramHoleSize[31:26]}+{0b,(DctSelBaseAddr[39:27]–{00h,DramHoleSize[31:27]})};
- **Case 4**: for any of the following conditions:
  - 1 DCT is enabled (F2x110[DctSelHiRngEn]=0)
  - 2 DCTs are enabled in channel interleaved mode and the same size (F2x110[DctSelHiRngEn]=0)

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

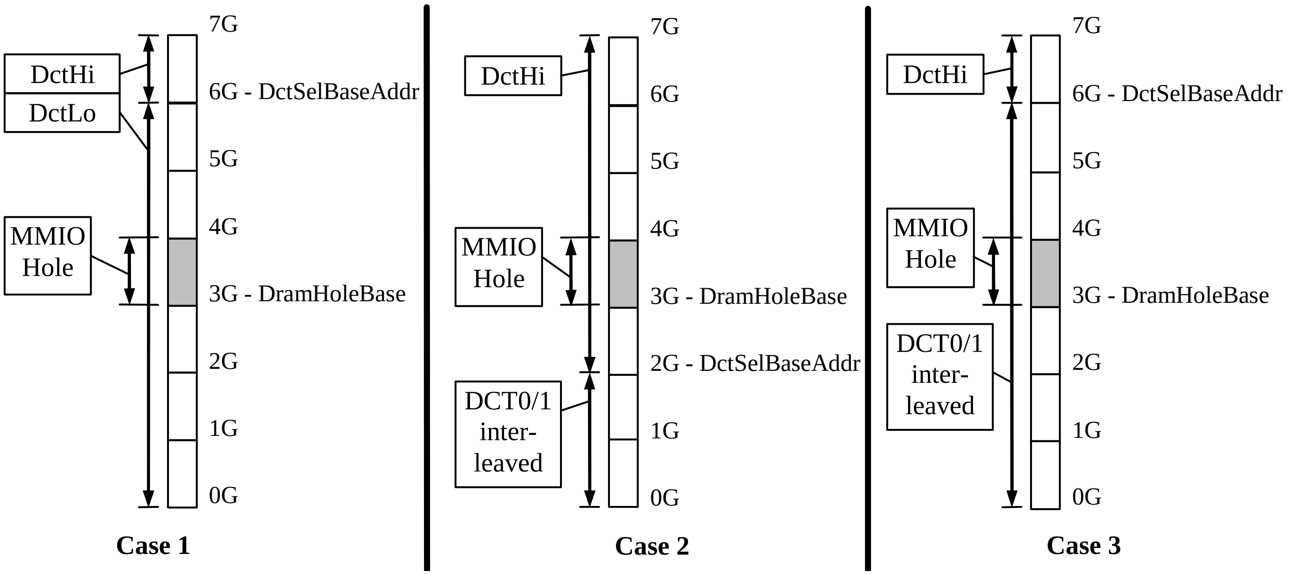

Figure 9: Example cases for programming DctSelBaseOffset

2.8.9 DRAM Thermal Protection (MEMHOT_L)

The DCT can throttle commands based on the state of the processor's MEMHOT_L pin. The MEMHOT_L pin is an asynchronous level sensitive input to the processor that, when asserted, indicates that a DRAM high temperature condition exists.

- The minimum assertion time for MEMHOT_L is 15 ns.
- The minimum deassertion time for MEMHOT_L is 15 ns.
- [The DRAM Controller Temperature Throttle Register] F2xA4 determines what actions are taken when MEMHOT_L is asserted.
- MEMHOT_L is ignored while:
  - PWROK is deasserted.
  - RESET_L is asserted.
- BIOS must ensure that throttling is disabled (F2xA4[ThrottleEn[1:0]]=00b) until DRAM training is complete.

JEDEC defines two DRAM device types: Standard (with a case temperature of 85 degrees C), and extended temperature (with a case temperature of 95 degrees C). The recommended usage and the interaction between DRAM case temperature and MEMHOT_L pin throttling is as follow:

- The BIOS may enable command throttling on a DRAM controller if the platform supports the MEMHOT_L pin by programming [The DRAM Controller Temperature Throttle Register] F2xA4.
  - The recommended usage is for this pin to be connected to one or more JEDEC defined on DIMM temperature sensor(s).
  - BIOS configures the temperature sensor(s) to assert MEMHOT_L pin active low when the trip point is exceeded and deassert MEMHOT_L when the temperature drops below the trip point minus the sensor defined hysteresis.
  - BIOS programs F2xA4[ThrottleEn] with the throttling mode to employ when the trip point has been exceeded.
- If all DIMMs support extended temperature range (specified in the DIMMs' SPD ROM), then BIOS programs the refresh rate to 3.9 us in F2x[1,0]8C[Tref].
- Standard and extended temperature devices may be mixed in a system. In this case, BIOS has two options:

  - BIOS programs the refresh rate to 3.9 us in F2x[1,0]8C[Tref], or,
  - BIOS configures the temperature sensor trip point for all DIMMs according to the 85 degrees C case temperature specification of the standard temperature DIMM(s).
- At startup, the BIOS determines if the DRAMs are too hot to enable a DCT, as determined by a call to a vendor routine, and delays for an amount of time to allow the devices to cool under the influence of the thermal solution. The vendor routine is recommended to check the temperature status of each DIMM.

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

2.8.10 DRAM Frequency

The actual DRAM frequency may be less than the DRAM frequency specified by F2x[1,0]94[MemClkFreq].
  - Table 19 on page 70 specifies the actual DRAM frequency as a function of the programmed DRAM frequency (F2x[1,0]94[MemClkFreq]) and the main PLL frequency (F3xD4[MainPllOpFreqIdEn, MainPllOpFreqId]).
  - The NB frequency is the same as the actual DRAM frequency.
  - In self refresh and if the memory clocks aren't tristated, then the DRAM frequency in self refresh can be modified by F2x[1,0]A0[PhyClkDivInSR].

Table 19: Actual DRAM frequency vs. PLL frequency

Main PLL Frequency	F2x[1,0]94[MemClkFreq]		
	266 MHz	**333 MHz**	**400 MHz**
800 MHz	266 MHz	320 MHz	400 MHz
900 MHz	257 MHz	300 MHz	360 MHz
1000 MHz	250 MHz	333 MHz	400 MHz
1100 MHz	244 MHz	314 MHz	367 MHz
1200 MHz	266 MHz	300 MHz	400 MHz
1300 MHz	260 MHz	325 MHz	371 MHz
1400 MHz	255 MHz	311 MHz	400 MHz
1500 MHz	250 MHz	333 MHz	375 MHz
1600 MHz	266 MHz	320 MHz	400 MHz
1700 MHz	262 MHz	309 MHz	378 MHz
1800 MHz	257 MHz	327 MHz	400 MHz
1900 MHz	253 MHz	317 MHz	380 MHz
2000 MHz	266 MHz	333 MHz	400 MHz
2100 MHz	263 MHz	323 MHz	382 MHz
2200 MHz	259 MHz	314 MHz	400 MHz
2300 MHz	256 MHz	329 MHz	383 MHz
2400 MHz	266 MHz	320 MHz	400 MHz
2500 MHz	263 MHz	333 MHz	385 MHz
2600 MHz	260 MHz	325 MHz	400 MHz
2700 MHz	257 MHz	318 MHz	386 MHz

Table 19: Actual DRAM frequency vs. PLL frequency

Main PLL Frequency	F2x[1,0]94[MemClkFreq]		
	266 MHz	**333 MHz**	**400 MHz**
2800 MHz	266 MHz	329 MHz	400 MHz
2900 MHz	264 MHz	322 MHz	387 MHz
3000 MHz	261 MHz	333 MHz	400 MHz
3100 MHz	258 MHz	326 MHz	388 MHz
3200 MHz	266 MHz	320 MHz	400 MHz

2.9 Core

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

2.9.1 Virtual Address Space

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

2.9.2 Number of Cores and Core Number

The processor supports 1 or 2 cores as follows:

- The number of cores supported by the processor is specified by F3xE8[CmpCap].
- The core number, *CpuCoreNum*, is provided to SW running on each core through CPUID Fn0000_0001_EBX[LocalApicId] and APIC20[ApicId]; CpuCoreNum also affects F0x68[Cpu1En]. CpuCoreNum, varies as the lowest integers from 0 to 1.
- The boot core is always the core reporting CpuCoreNum = 0.

2.9.3 Access Type Determination

The access type determination and destination affects routing specified in section 2.6.3 [Northbridge Routing].

2.9.3.1 Memory Access to the Physical Address Space

All memory accesses to the physical address space from a core are sent to the 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.

The above is configured by 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. (Highest priority) TSEG & ASEG (see MSRC001_0112 and MSRC001_0113).

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

1. The CPU compares the address against [The Top Of Memory Register (TOP_MEM)] MSRC001_001A, and [The Top Of Memory 2 Register (TOM2)] MSRC001_001D, to determine if the default access destination is DRAM or MMIO space.
2. For addresses below 1M byte, the address is then compared against the appropriate Fixed MTRRs to override the default access destination. Each fixed MTRR includes two bits, RdDram and WrDram, that determine the destination based on the access type. See MSR0000_02[6F:68, 59, 58, 50].
3. The CPU then compares the address against the IORRs (MSRC001_00[18, 16] and MSRC001_00[19, 17]); if it matches, the default access destination is overridden as specified by the IORRs. BIOS can use the IORRs to create an IO hole within a range of addresses that would normally be mapped to DRAM. It can also use the IORRs to re-assert a DRAM destination for a range of addresses that fall within a bigger IO hole that overlays DRAM. 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 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.

The above is configured by BIOS and does not require any setup or changes by system software.

Note: BIOS must ensure that when it makes IO cacheable, IO devices in the cacheable region will respond correctly to cacheable requests. If this requirement cannot be met, BIOS must protect these IO regions from cacheable requests. The recommended method is to make them not cacheable.

2.9.4 Timers

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

- [The Time Stamp Counter Register (TSC)] MSR0000_0010; the TSC increments at the rate specified by MSRC001_0015[TscFreqSel].
- The APIC timer (APIC380 and APIC390), which 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 APIC ID Enumeration Requirements

System hardware and BIOS must ensure that the number of cores per processor (CPUID Fn8000_0008_ECX[NC]) exposed to the operating system by all tables, registers, and instructions across all cores in the system is identical. See2.16.1 [Multi-Core Support] to derive NC.

Operating systems are expected to assign APIC20[ApicId] values as described below:

ApicId[core0] = 00h.
ApicId[core1] = 01h.

2.10 Thermal Functions

Thermal functions HTC and THERMTRIP are intended to maintain the processor's 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.
- Sending processor to the THERMTRIP state to prevent damage.

The processor thermal-related circuitry includes (1) the therm sense macro (TSM) for determining the temperature of the processor and (2) logic that uses the temperature from the TSM. The processor includes a thermal diode connected to pins as well.

2.10.1 The Tctl Temperature Scale

Tctl is the processor temperature control value, used by the platform to control its 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 Sideband Temperature Sensor Interface (SB-TSI)

The SB-TSI is used by an external SMBus master to access the internal temperature sensor and to specify temperature thresholds. The processor has access to the SB-TSI registers through [The SBI Address Register] F3x1E8 and [The SBI Data Register] F3x1EC. See SBI Temperature Sensor Interface (SB-TSI) Specification, #40821. 100 KHz standard-mode and 400 KHz fast-mode are supported. 3.4 MHz high-speed mode is not supported.

2.10.4 Temperature-Driven Logic

The temperature calculated by the TSM is used by HTC, THERMTRIP, the PROCHOT signal, and SB-TSI.

2.10.4.1 Hardware Thermal Control (HTC) and PROCHOT_L

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]. While in the HTC-active state, software should not change the following: All F3x64 fields (except for HtcActSts and HtcEn), MSRC001_001F[DisProcHotPin]. Any change to the previous list of fields when in the HTC-active state can result in undefined behavior. HTC status and control is provided through 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.

2.10.4.2 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 during cold reset (shortly after PWROK asserts, before RESET_L deasserts); it remains enabled in all other processor states, including during warm reset (while RESET_L is asserted). The THER-

MTRIP 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 on all planes.
- In addition, the IO Hub 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 both BCS and ECS configuration space registers located at bus 0 device 24. See 2.11.3 [Processor Configuration Space] for more information on processor configuration space.

Configuration space is accessed by the processor through two methods:

- IO-space configuration: IO instructions to addresses CF8h and CFCh.
  - Software restricts IO-space configuration to 1 core at a time.
  - Enabled through [The IO-Space Configuration Address Register] IOCF8[ConfigEn], which allows access to BCS.
  - Access to ECS enabled through [The Northbridge Configuration Register (NB_CFG)] MSRC001_001F[EnableCf8ExtCfg].
  - Use of IO-space configuration can be programmed to generate GP faults through [The Hardware Configuration Register (HWCR)] MSRC001_0015[IoCfgGpFault].
  - SMI trapping for these accesses is specified by [The IO Trap Control Register (SMI_ON_IO_TRAP_CTL_STS)] MSRC001_0054 and [The IO Trap Registers (SMI_ON_IO_TRAP_[3:0])] MSRC001_00[53:50].
- MMIO configuration: configuration space is a region of memory space.
  - The base address and size of this range is specified by [The MMIO Configuration Base Address 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 configuration space as follows:
    - Address[31:0] = {0000b, bus[7:0], device[4:0], function[2:0], offset[11:0]}.
  - SMI trapping for MMIO configuration accesses is not supported.

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.

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 writes may result in unexpected behavior.

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

2.11.3 Processor Configuration Space

The processor includes configuration space as described in section 3 [Registers]. Accesses to unimplemented registers of implemented functions are ignored: writes dropped; reads return 0’s. Accesses to unimplemented functions are also ignored: writes are dropped; 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 are routed to the compatibility (subtractive) address space. If such requests are master aborted, then the processor can log the event. Configuration-space transactions from the link are master aborted.

2.12 Debug Support

2.12.1 . Built In Self Test (BIST)

The EAX register and [The BIST Results Register] MSRC001_0060 provide BIST results after each reset. The results in EAX are identical to and defined by MSRC001_0060. A value of 0 indicates that no BIST failures were detected.

2.13 RAS: Reliability, Availability, and Serviceability

This section applies reliability, availability, and serviceability, or *RAS*.

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 NB.

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].

2.13.1.1 Machine Check Registers

The presence of the machine check registers is indicated by CPUID Fn0000_0001_EDX[MCA]. The ability of hardware to generate a machine check exception upon an error is indicated by CPUID Fn0000_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 5-register-bank matrix.
  - The four register types are:
    - **MCi_CTL**, The Machine Check Control Registers: MSR0000_0400, MSR0000_0404, MSR0000_0408, MSR0000_040C, MSR0000_0410.
    - **MCi_STATUS**: The Machine Check Status Registers: MSR0000_0401, MSR0000_0405, MSR0000_0409, MSR0000_040D, MSR0000_0411.
    - **MCi_ADDR**: The Machine Check Address Registers: MSR0000_0402, MSR0000_0406, MSR0000_040A, MSR0000_040E, MSR0000_0412.
    - **MCi_MISC**: The Machine Check Miscellaneous Registers: MSR0000_0403, MSR0000_0407, MSR0000_040B, MSR0000_040F, MSR0000_0413.
  - The 5 error-reporting register banks supported are (Where x=[MISC, ADDR, STATUS, CTL]):
    - **MC0_x, DC**: MSR0000_04[03:00], data cache machine check registers.
    - **MC1_x, IC**: MSR0000_04[07:04], instruction cache machine check registers.
    - **MC2_x, BU**: MSR0000_04[0B:08], bus unit machine check registers.
    - **MC3_x, LS**: MSR0000_04[0F:0C], load-store machine check registers.
    - **MC4_x, NB**: MSR0000_04[13:10], NB machine check registers. The NB MC registers are accessible from configuration space as well.

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.

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.

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.

MC[3:0]_CTL_MASK (MSRC001_00[47:44]) allows BIOS to masks error reporting, but does not mask error detection. MC4_CTL_MASK (MSRC001_0048) allows BIOS to mask error detection. The masking of error detection prevents error responses. See MSRC001_00[48:44].

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:

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.
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.

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.
- MC[3:0]: MC[3:0]_CTL_MASK has no effect on error detection and logging.
  MC4: The corresponding MC4 mask bit for the error in MC4_CTL_MASK is cleared to 0. See [The Machine Check Control Mask Registers (MCi_CTL_MASK)] MSRC001_00[48:44].

An error is considered enabled for reporting if:
- MC[3:0]: The corresponding MC[3:0] mask bit for the error in MC[3:0]_CTL_MASK is cleared to 0. See [The Machine Check Control Mask Registers (MCi_CTL_MASK)] MSRC001_00[48:44].
  MC4: 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 20 indicates which errors are overwritten in the error status registers for all MCA banks (MC0 to MC4).

Table 20: 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. It is possible 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 ensured 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.

2.14 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. 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]. The corresponding bit in [The Trigger Mode Registers] APIC[1F0:180] is set if the interrupt is level-triggered and cleared if edge-triggered.

When the processor has completed service for an interrupt, it performs a write to 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 APIC90 [Arbitration Priority Register], 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 effect 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 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).

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 go to the IO Hub and trigger a global SMI interrupt message back to the all cores.

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

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

The status for these is stored in SMMFEC4.

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 21: 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

Table 21: SMM initial state

Register	SMM Initial State
CR4	0000_0000_0000_0000h
GDTR	Unmodified
LDTR	Unmodified
IDTR	Unmodified
TR	Unmodified
DR6	Unmodified
DR7	0000_0000_0000_0400h
EFER	All bits are cleared except bit 12 (SVME) which is unmodified.

2.14.2.5 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 22: 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 (see note 1)	
FE48h	Quadword		Descriptor in memory format	
FE50h	Word	GS	Selector	Read-only
FE52h	2 Bytes		reserved	
FE54h	Doubleword		GS Base (see note 1)	
FE58h	Quadword		Descriptor in memory format	
FE60h	4 Bytes	GDTR	reserved	Read-only
FE64h	Word		Limit	
FE66h	2 Bytes		reserved	
FE68h	Quadword		Descriptor in memory format	

Table 22: 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	[The SMM SVM State] SMMFED8		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 22: 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: All addresses are stored in canonical form.

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 a 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
31:17	Reserved.
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 MSRC001_0055[IntrPndMsg] (when that bit is high).
9:4	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 re-fetched 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 that 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=Reserved.

SMMFEFC SMM-Revision Identifier

SMM entry state: 0003_0064h.

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

SMMFF00 SMM Base Address Register (SMM_BASE)

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_L 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.

2.14.2.10 Multiple Unsynchronized SMI Sources

When more than one IO device in the system is enabled to signal an SMI, or when a single device may signal multiple SMI messages without hardware synchronization (e.g. using an end of SMI gate), the processor cores may enter a state where all cores' SMI interrupt pending status bits do not match. As a result, the application processor cores, which are usually slaved by the boot strap core in handling SMIs and controlling SMM flow, may enter a software loop in SMM.

The BIOS must take special care to ensure that all cores have entered SMM prior to accessing shared I/O resources and all processors' SMI interrupt status bits are synchronized. The act of synchronizing cores into SMM is called spring boarding. SMI spring boarding applies to all multi-core processors that are affected by the platform architectural factors mentioned above.

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

The software requirements for the suggested SMI spring boarding are as follows.

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

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

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

```
//Example:
InLineASM{
    BTS CheckSpringBoard,0    ;Try to obtain ownership of semaphore
    JC Step_2:
    CALL CheckIOHUB_SMIEVT  ;proc returns ZF=1 for no events
    JNZ Step_2:
    CALL Do_SpringBoard     ;Trigger SMI and then RSM
Step_2:
}
```

2. Decrement the NotInSMM variable. Wait for NotInSMM=0. See Note 1.
3. Execute the core-local event SMI handler. Using a third semaphore (not described here), synchronize processor core execution at the end of the task. After all processor cores have executed, proceed to step 4. The following is a brief description of the task for each processor core:
   - Check all enabled processor-core-local SMI status bits in the core’s private or MSR address space. Handle the event if possible, or pass information necessary to handle the event to a mailbox for the boot strap processor to handle.
   - An exclusive mailbox must exist for each processor core for each core local event.
   - On-line spare events should be handled in this task by the individual core for optimal performance. Assign one core of a dual core processor to handle On-line spare. These events may be optionally handled by the BSC just as other global events.
   - Wait for all processor cores to complete this task at least once.
4. If the current processor core executing instructions is not the BSC then jump to step 5. If the core executing instructions is the BSC then jump to the modified main SMI handler task, described below.
   - Check all enabled SMI status bits in the I/O hub. Check mailboxes for event status.
   - For each event, handle the event and clear the corresponding status bit.
   - Repeat until all enabled SMI status bits are clear and no mailbox events remain.
   - Set NotInSMM=NumCPUs. (Jump to step 5.)
5. Decrement the WaitInSMM variable. Wait for WaitInSMM=0. See Note 2.
6. Increment the WaitInSMM variable. Wait for WaitInSMM=NumCPUs.
7. If the current processor core executing instructions is the BSC then reset CheckSpringBoard to zero.
8. Resume from SMM with the RSM instruction.

Notes:

1. To support a secure startup by the secure loader the BIOS must provide a timeout escape from the otherwise endless loop. The timeout value should be large enough to account for the latency of all processor cores entering SMM. The maximum SMM entrance latency is defined by the platform’s I/O sub-system, not the processor. AMD recommends a value of twice the watchdog timer count. See [The MCA NB Configuration Register] F3x44 for more information on the watchdog time-out value.

If a time-out occurs in the wait loop, the BIOS (the last core to decrement NotInSMM) should record the number of cores that have not entered SMM and all cores must fall out of the loop.

2. If a time-out occurs in the wait loop in step 2, the BIOS must not wait for WaitInSMM=0. Instead it must wait for WaitInSMM=”the number of cores recorded in step 2”.

2.15 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

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

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

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

- Disable AMD Virtualization™, with a user supplied key.
    - MSRC001_0114[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] 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[NC]. The legacy method utilizes the CPUID Fn0000_0001_EBX[LogicalProcessorCount].

2.17 Performance Monitoring

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].

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.
- **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 40. See 2.11 [Configuration Space], for details about configuration space. Each processor includes five functions, 0 through 4.
- **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 [The CPUID Instruction Registers] 3.9.
- **MSRXXXX_XXXX**: model specific registers; XXXX_XXXX is the MSR number. This space is accessed through x86-defined RDMSR and WRMSR instructions.

The register number can use “[]” notation as defined in “Arithmetic and Logical Operators” on page 13. For example, F2x[1,0][4C:40] is a shorthand notation for F2x040, F2x044, F2x048, F2x04C, F2x140, F2x144, F2x148, and F2x14C.

The processor includes a single set of IO-space and configuration-space registers. However, APIC, CPUID, and MSR register spaces are implemented once per processor core. 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 23: Terminology in register descriptions

Terminology	Description
BIOS:	The recommended value to be set by software, either BIOS, OS, or driver. The format is defined to be BIOS:<integer-expression> [if <conditional-expression>]. The brackets indicate that the conditional expression is optional. If the conditional expression is absent then the software recommendation is always the integer-expression value. • If a software recommendation does not exist for all conditions or for all bits of a condition, then software is recommended not to change the value of the specified bit(s). • If “BIOS:” occurs in a register field: • The recommended value is applied to the field. • If “BIOS:” occurs after a register name but outside of a register field table row: • The recommended value is applied to the width of the register. • E.g.: BIOS: 4h.
Read or read-only	Capable of being read by software. Read-only implies that the register cannot be written by software.

Table 23: Terminology in register descriptions

Terminology	Description
Write	Capable of being written by software.
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.
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-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; 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

The MSRs that control NB functions are shared between all cores in a multi-core processor. If control of NB 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. The MSRs that control NB functions are listed as follows:

- MSR0000_0410
- MSR0000_0411
- MSR0000_0412
- MSRC001_001F
- MSRC001_00[48:44]
- MSRC001_0058

3.2 IO Space Registers

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

IOCF8 IO-Space Configuration Address Register

Reset: 0000 0000h.

[The IO-Space Configuration Address Register] IOCF8, and [The IO-Space Configuration Data Port] IOCFC, are used to access system configuration space, as defined by the PCI specification. IOCF8 provides the address register and IOCFC provides the data port. Software sets up the configuration address by writing to IOCF8. Then, when an access is made to IOCFC, the processor generates the corresponding configuration access to the address specified in IOCF8. See also section 2.11 [Configuration Space].

IOCF8 may only be accessed through aligned, DW IO reads and writes; otherwise, the accesses are passed to the appropriate IO link. Accesses to IOCF8 and IOCFC received from an IO link are treated as all other IO transactions received from an IO link and are forwarded based on the settings in [The IO-Space Base/Limit Registers] F1x[C4,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 to IOCFC 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

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

3.3 Function 0 Link Configuration Registers

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

F0x00 Device/Vendor ID Register

Reset: 1300 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 0040h.

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 0080h.

Bits	Description
31:8	Reserved.
7:0	**CapPtr: capabilities pointer.** Read-only: 80h. Specifies the offset of the link capabilities block.

F0x60 Node ID Register

Reset: 0000 0000h.

Bits	Description

31:7	Reserved.
6:4	**NodeCnt[2:0]: node count bits[2:0]**. Read-only. This specifies the number of coherent nodes in the system.
3	Reserved.
2:0	**NodeId[2:0]: node ID bits[2:0]**. Read-only. This specifies the node ID of the processor node.

F0x64 Unit ID Register

Reset: 0000 0000h.

Bits	Description
31:11	Reserved.
10:8	**SbLink: Southbridge (IO Hub) link ID.** Read-only. This field specifies the link to which the system IO Hub is connected.
7:0	Reserved.

F0x68 Link Transaction Control Register

Reset: 0000 0000h, except bit 24.

Bits	Description
31:25	Reserved.
24	• **DispRefModeEn**. Read-write; changes take effect on next warm reset. Cold reset: 0. 1=Enables support for display-refresh ordering rules. See section 2.6.3.2.1 [Display Refresh And IFCM].
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) or peer-to-peer, which may be outstanding on the link. 00b = no limit. 01b = limited to 1. 10b = limited to 4. 11b = limited to 8.
20	**DisSeqIdReqUID: disable using requester UID for SeqID**. Read-write. 0=SeqID is assigned the UID of the requesting CPU, such that CPU0=2h and CPU1=3h. 1=SeqID of 0h is used.
19	**ApicExtSpur: APIC extended spurious vector enable.** Read-write. This enables the extended APIC spurious vector functionality. 0=The lower 4 bits of the spurious vector are read-only 1111b. 1=The lower 4 bits of the spurious vector are writable.
18	**ApicExtId: APIC extended ID enable.** Read-write. 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. 0=ExtInt/NMI interrupts delivered unchanged.
15	RAZ.

14:13	**BufRelPri: buffer release priority select.** Read-write. BIOS: 10b. 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.
12	Reserved.
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.
10:8	Reserved.
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: core 1 enable**. Read-write. This bit is used to enable the CPU1 core after a reset. 1=Enable core 1 to start fetching and executing code from the boot vector. Note: SW must not try to assert Cpu1En if CpuCoreNum=0, which indicates that only CPU0 is present; see CpuCoreNum in section [The Number of Cores and Core Number] 2.9.2.
4:0	Reserved.

F0x6C Link Initialization Control Register

Reset: 0000 ???0h; see individual bit definitions for reset details.

Bits	Description
31:11	Reserved.
10:9	**BiosRstDet[2:1]: BIOS reset detect bits[2:1].** Read-write. Cold reset: 0. See bit[5] of this register.
8:7	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 BiosRstDet[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.
3:1	Reserved.
0	RAZ.

F0x80 Link Capabilities Register

This register is derived from link register specifications.

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. 00h.
7:0	**CapID: capabilities ID.** Read-only. Reset: 08h. Indicates link technology capability.

F0x84 Link Control Register

This register is derived from link register specifications.

The reset value of this register is the earliest point that this register can be read by software, which is when the link initialization is successfully completed, or 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.

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 — 101b — 4 bits 000b — 8 bits — 100b — 2 bits 111b — not connected The cold reset value of this field depends on the width of the link of the connecting device, per the link specification. Note: after this field is written to by software, the link width does not change until either a warm reset or LDTSTOP disconnect. This field must not be modified after link power management is enabled.
27	Reserved.
26:24	**WidthIn: link width in.** Read-write. Cold reset: (See text below). Specifies the operating width of the incoming link. See F0x84[WidthOut] for legal values. The cold reset value of this field depends on the width of the link of the connecting device, per the link specification. Note: after this field is written to by software, the link width does not change until either a warm reset or LDTSTOP disconnect. This field must not be modified after link power management is enabled.
23	Reserved.
22:20	**MaxWidthOut: max link width out.** Read-only. Reset: 001b (16 bits). This specifies the width of the outgoing link to be 8 bits or 16 bits wide, depending on the processor version. See F0x84[WidthOut] for the encoding.
19	Reserved.

18:16	**MaxWidthIn: max link width in.** Read-only. Reset: 001b (16 bits). This specifies the width of the incoming link to be 8 bits or 16 bits wide, depending on the processor version. See F0x84[WidthOut] for the encoding.
15	Reserved.
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 us. This bit is ignored at Gen3 frequencies.
13	**LdtStopTriEn: LDTSTOP Tristate Enable.** Read-write. Cold reset: 0. BIOS: 1. 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. This bit does not apply to inactive lanes when F0x170[TxInLnSt] (F0x170[RxInLnSt] for receive lanes) is set to 11b. This bit is ignored by hardware when the link is operating at Gen3 frequencies.
12	**IsocEn: isochronous flow-control mode enable.** Read-write; changes take effect on next warm reset. Cold reset: 0. This bit is set to place the link into isochronous flow-control mode (IFCM), as defined by the link specification. 1=IFCM. 0=Normal flow-control mode (NFCM). See section 2.6.3.2.1 [Display Refresh And IFCM].
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: • CRC error in Gen1 mode is detected on the link (if enabled by CrcFloodEn). • Sync flood received. • The link fails to connect in Gen3.
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=Detected CRC errors result in sync packets to the enabled outgoing link and the F0x84[LinkFail] bit is set. 0=CRC errors do not result in sync packets or setting the F0x84[LinkFail] bit. In Gen3 protocol, exceeding the F0x150[TotalRetryAttempts] limit results in a sync flood and setting the LinkFail bit regardless of how CrcFloodEn is set. The resulting sync flood does not propagate beyond the link unless CrcFloodEn is set. This bit is ignored if F3x44[SyncPktGenDis] is set.
0	Reserved.

F0x88 Link Frequency/Revision Register

This register is derived from link register specifications.

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	**CtlTimeout: CTL Timeou**t. Read-write. Reset: 0. Indicates how long CTL may be low before a protocol error is logged in the MCA. 0=1 ms. 1=1 second. BIOS is recommended to not change this bit.
14:12	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 LDTSTOP disconnect. A read to this field returns that last written value. A write to this field of an unsupported frequency, as indicated by F0x88[LnkFreqCap], may produce undefined behavior. This field must not be modified after link power management is enabled.
7:0	**Revision.** Read-only, 60h. Indicates that the processor is designed to version 3.00 of the link specification.

F0x8C Link Feature Capability Register

This register is derived from link register specifications. 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, 0.
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.

F0x90 Link Base Channel Buffer Count Register

Cold reset: 0021_218Ch. Read-write; changes take effect on next warm reset. See section 2.6.3.2.1 [Display Refresh And IFCM].

Bits	Description
31:24	Reserved.
23:20	**RspData: response data buffer count**. Cold reset: 02h. (2 buffers)
19	Reserved.
18:15	**NpReqData: non-posted request data buffer count**. Cold reset: 02h. (2 buffers)
14:10	**RspCmd: response command buffer count**. Cold reset: 08h. (8 buffers)
9:5	**PReq: posted request command and data buffer count**. Cold reset: 0Ch. (12 buffers) This specifies the number of posted command and posted data buffers allocated.
4:0	**NpReqCmd: non-posted request command buffer count**. Cold reset: 0Ch. (12 buffers)

F0x94 Link Isochronous Channel Buffer Count Register

Cold reset: 0000 0000h. Read-write; changes take effect on next warm reset. See section 2.6.3.2.1 [Display Refresh And IFCM].

Bits	Description
31:24	Reserved.
23:20	**IsocRspData: isochronous response data buffer count**.
19	Reserved.
18:15	**IsocNpReqData: isochronous non-posted request data buffer count**.
14:10	**IsocRspCmd: isochronous response command buffer count**.
9	Reserved.

8:5	**IsocPReq: isochronous posted request command and data buffer count**. This specifies the number of isochronous posted command and posted data buffers allocated.
4:0	**IsocNpReqCmd: isochronous non-posted request command buffer count**.

F0x98 Link Type Register

Reset: 0000 000?h.

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. Reset: 1. This bit specifies that the link type is an IO link.
1	**InitComplete: initialization complete.** Read-only. Cold reset: 0. 1=Link initialization is complete. This is a duplicate of [The Link Control Register] F0x84[InitComplete]. The NC bit is invalid until link initialization is complete.
0	**LinkCon: link connected.** Read-only. Reset: 1. 1=The link is connected to another device. 0=The link is not connected. This is not valid until LinkConPend=0.

F0x130 Link Retry Register

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 processor.
15:12	Reserved.
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 an Gen3 link defined 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 only incremented when a retry is initiated by the processor due to the detection of a retry-able error or a training timeout. Retries initiated by another device have no effect on this counter. This field is ignored if when operating at Gen1 link frequencies. • BIOS is recommended to leave this field in the reset state.
5:4	Reserved.

3	**DisRetryDataError: disable link retry on data packet error**. Read-write. Reset: 0. 1=The processor 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 processor 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.

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 F0x130[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 errorse.g.; if TotalRetryAttempts=0, then the link is determined to have failed as a result of the 1 error. The retry attempt counter for a link is incremented each time F0x130[RetryCount] for that link is incremented.
15:7	Reserved.
6:5	**ForceErrType: force error type**. Read-write. Specifies the error type generated by F0x130[ForceRetryError] and F0x84[CrcForceErr]. 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). 1xb Forces per-packet CRC error on a data packet only.
4	**MultRetryErr: multiple retry force error**. Read-write. 1=Inhibits hardware clearing of [The Link Retry Register] F0x130[ForceRetryError], thereby causing multiple link retry errors. This can be used to test software associated with reporting of multiple link reconnect failures.
3:0	Reserved.

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 processor 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 processor, and there is no data that follows, then the processor sends at least 2 DWORDs of NOPs (possibly including buffer release credits) before generating the next command packet. This bit applies to Gen3 frequencies only. Some IO devices may require this bit to be clear. 1=The processor does not limit the rate at which commands are generated.
9:0	Reserved.

F0x16C Link Global Extended Control Registers

Cold reset: 0074_003Ah. Further information about these bits can be found in the Gen3 link specification.

Bits	Description
31:23	Reserved.
22:17	**FullT0Time: full training 0 time**. Read-write. Cold reset: 3Ah. IF (F4x184_x[530A, 520A][DllLockFastModeEn]) THEN BIOS: 2Ah.ELSE BIOS: 3Ah.ENDIF. This specifies the amount of time to spend in training 0 following a warm reset, frequency change, or when the full T0 training period is invoked due to expiration of the idle timer, as described in F0x16C[ForceFullT0]. Encodings are the same as T0Time. BIOS should set FullT0Time according to the maximum T0 training time requirement for the link’s far-side receiver phase recovery time as determined by characterization.
16	**ImmUpdate: immediate update**. Read-write. Cold reset: 0. Many of the link phy registers, accessed through F4x180, 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 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 disconnect or warm reset. 1=Writes to the link phy registers are passed to the phy immediately. In either case, reads always return the current active value, which is not necessarily the last value written; it may be the value written before the last link disconnect, or generated by the LMM override mechanism. Note: The LMM override of phy fields by F4x174_x[0F:00] ignores F0x16C[ImmUpdate] and always takes place when the link is disconnected.

15:13	**ForceFullT0: force full T0 training time**. Read-write. Cold reset: 000b. BIOS: 110b. This specifies the period of time that lanes can be in power-savings states (LS1, LS2, or Warm Reset for inactive lanes) 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. This field must not be modified after link power management is enabled. Note that inactive lanes may be disconnected longer than active lanes, and may force the whole link to undergo a full T0 training when they are reactivated. This may be avoided by performing inactive lane refresh prior to reactivating them. Once any link defined CLMC command has been received, only inactive lanes below the width programmed in link F0x84[WidthIn/Out] are monitored. If less than the specified time has expired, then training 0 specified by F0x16C[T0Time] is used, otherwise the Training 0 time specified by F0x16C[FullT0Time] is used. The bits are encoded as follows: 000b = Always use F0x16C[T0Time]. 100b = 3.2 ms. 001b = 400 us. 101b = 6.4 ms. 010b = 800 us. 110b = 12.8 ms. 011b = 1.6 ms. 111b = 25.6 ms.
12:6	Reserved.
5:0	**T0Time: training 0 time**. Read-write. Cold reset: 3Ah. BIOS: 26h. (12us) Specifies the amount of time to spend in training 0 when exiting the disconnected state. See also F0x16C[ForceFullT0], F4x184_x[530A, 520A][Ls2ExitTime], and section 2.7.5 [Link LDTSTOP_L Disconnect-Reconnect]. If T0Time[5:4]=00b, then the time = T0Time[3:0] * 0.1 us (ranging from 0.0 to 1.5 us). If T0Time[5:4]=01b, then the time = T0Time[3:0] * 0.5 us (ranging from 0.0 to 7.5 us). If T0Time[5:4]=10b, then the time = T0Time[3:0] * 2.0 us (ranging from 0.0 to 30 us). If T0Time[5:4]=11b and T0Time[3:0] ranges from 0h to Ah, then the time = T0Time[3:0] * 20 us (ranging from 0.0 to 200 us). 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 link's far-side receiver phase recovery time when it has been disconnected for a period not exceeding the time specified by F0x16C[ForceFullT0], as determined by characterization. The programmed time may be exceeded by 10% or 100ns, whichever is greater. This field must not be modified after link power management is enabled.

F0x170 Link Extended Control Registers

Cold reset: 0000_0001h. Further information about these bits can be found in the Gen3 link specification.

Bits	Description
31:27	Reserved.
26:25	**TxInLnSt: transmit inactive lane state**. Read-write. Cold reset: 00b. Specifies the state of inactive transmit lanes of the link at all frequencies. Updates to this field do not take effect until a warm reset or LDTSTOP disconnect. This field must not be modified after link power management is enabled. This field controls all transmit lanes beyond the width programmed in F0x84[WidthOut, WidthIn]. Bits State Used 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. 11b Same as disconnected per F0x170[TxLSSel].

24:23	**RxInLnSt: receive inactive lane state**. Read-write. Cold reset: 00b. Specifies the state of inactive receive lanes of the link at all frequencies. Updates to this field do not take effect until a warm reset or LDTSTOP disconnect. This field must not be modified after link power management is enabled. This field controls all receive lanes beyond the width programmed in F0x84[WidthOut, WidthIn]. Bits State Used 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. 11b Same as disconnected per F0x170[RxLSSel].
22:21	**TxLSSel: transmit LDTSTOP mode select**. Read-write. Cold reset: 00b. BIOS: Configured to be the highest power savings mode supported by link DS device receiver. This field specifies the state used on the transmitter for power reduction when the link is disconnected and for inactive lanes when F0x170[TxInLnSt]==11b. Updates to this field do not take effect until a warm reset or LDTSTOP disconnect. This field must not be modified after link power management is enabled. Bits State Used 00b LS1 01b LS0 10b LS2 11b Reserved.
20:14	Reserved.
13:12	**LaneSel: lanes select**. Read-write. Cold reset: 00b. This field specifies how receive (RX) lanes are translated into transmit (TX) lanes when the link is 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 {CAD[12:8], CTL0, CAD[7:0], CTL1, CAD[15:13]} {CAD[6:0], CTL0, CAD[7]} 10b {CTL0, CAD[7:0], CTL1, CAD[15:8]} {CAD[4:0], CTL0, CAD[7:5]} 11b {CAD[4:0], CTL1, CAD[15:8], CTL0, CAD[7:5]} {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 {CAD[3:0], CTL0} {CAD[1:0], CTL0} 10b {CAD[2:0], CTL0, CAD[3]} {CAD[0], CTL0, CAD[1]} 11b {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 for a 16-bit link, LaneSel[1] selects which sublink is received by the BIST engine. 0=sublink 0, 1=sublink 1. • In BIST mode for 2-bit, 4-bit, and 8-bit links, LaneSel[1:0] performs the lane re-mapping described above for ILM mode on the receiver prior to descrambling the incoming data.
11	**ILMEn: internal loopback mode (ILM) enable**. Read-write. Cold reset: 0. Warm reset: see following text. 1=Enables ILM for receiver and transmitter upon the next LDTSTOP Disconnect or warm reset. Cleared by hardware upon the subsequent LDTSTOP Disconnect or warm reset. F4x184_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. Updates to this field do not take effect until a warm reset. 1=The link BIST engine is enabled. LDTSTOP must not be asserted when enabling or disabling BIST.

9	Reserved.
8:7	**RxLSSel: receive LDTSTOP mode select**. Read-write. Cold reset: 00b. BIOS: Configured to be the highest power savings mode supported by link DS device transmitter. This field specifies the state used in the receiver for power reduction when the link is disconnected and for inactive lanes when F0x170[RxInLnSt]==11b. Updates to this field do not take effect until a warm reset or LDTSTOP disconnect. This field must not be modified after link power management is enabled. Bits State Used 00b LS1 01b LS0 10b LS2 11b Reserved
6:4	Reserved.
3	**ScrambleEn: scrambling enable**. Read-write. Cold reset: 0. Should be cleared to 0 when F0x88[Freq] is programmed to a Gen1 frequency, and set to 1 when F0x88[Freq] is programmed to a Gen3 frequency. It may be left at 0 for Gen3 frequencies for testing purposes. Updates to this bit take effect on warm reset and LDTSTOP disconnect.
2:1	Reserved.
0	**Ganged**. Read-only. Reset: 1. 1=The link is ganged.

F0x1A4 Downstream ONION Buffer Count Register

Read-write; changes take effect on next warm reset. See section 2.6.3.2.1 [Display Refresh And IFCM].

Bits	Description
31	Reserved.
30:26	**DnHiRespBC: downstream high priority response buffer count**. Cold reset: 0h.
25:21	**DnHiNpreqBC: downstream high priority non-posted request buffer count**. Cold reset: 0h.
20:16	**DnHiPreqBC: downstream high priority posted request buffer count**. Cold reset: 0h.
15	Reserved.
14:10	**DnLoRespBC: downstream low priority response buffer count**. Cold reset: Fh.
9:5	**DnLoNpreqBC: downstream low priority non-posted request buffer count**. Cold reset: 4h.
4:0	**DnLoPreqBC: downstream low priority posted request buffer count**. Cold reset: 6h.

F0x1D0 Extended Link Buffer Count Register

See section 2.6.3.2.1 [Display Refresh And IFCM].

Bits	Description
31:11	Reserved.
10	**TxClkHysMuxEn: slow clocks to HT phy to avoid hysteresis**. Read-Write. Cold reset: 0. BIOS: 1. 1=The transmit clocks from the HTG to the HT phy are slowed rather than stopped.
9	**PllTreeDisOnDiscEn: PLL clock tree disable on disconnect enable**. Read-Write. Cold reset: 0. BIOS: 0. 1=Enable PLL clock tree to be disabled during link disconnect when both receiver and transmitter are in LS2, for power savings.

8	**PhyClkDisOnDiscEn: phy clock disable on disconnect enable**. Read-Write. Cold reset: 0. BIOS: 0. 1=Enable high-speed clock input from the phy to be disabled during link disconnect, for power savings.
7:6	Reserved.
5	**HiPriModeEn: high priority mode enable**. Read-write; changes take effect on next warm reset. Cold reset: 0. See section 2.6.3.2.1 [Display Refresh And IFCM].
4:0	Reserved.

F0x1D4 Downstream ONION Buffer Count Register 2

Cold reset: 0000_0000h. Read-write; changes take effect on next warm reset. See section 2.6.3.2.1 [Display Refresh And IFCM].

Bits	Description
31:10	Reserved.
9:5	**DnDRRespBC : display refresh response buffer count**. Cold reset: 0.
4:0	**DnPoolBC : downstream pool buffer count**. Cold reset: 0.

3.4 Function 1 Address Map Registers

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

F1x00 Device/Vendor ID Register

Reset: 1301 1022h.

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

F1x04 Status/Command Register

Reset: 0000 0000h. Read-only.

Bits	Description
31:16	**Status.**
15:0	**Command.**

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.

F1x34 Capabilities Pointer Register

Reset: 0000 0000h.

Bits	Description
31:8	Reserved.
7:0	**CapPtr: capabilities pointer.** Read-only, 00h.

F1x[44,40] DRAM Base/Limit Registers

These registers specify the DRAM address range, base and limit:

Base Address	Limit Address
F1x40	F1x44

DRAM mapping rules:

- Transaction addresses are within the defined range if:
  {DramBase[39:24], 00_0000h} <= address[39:0] <= {DramLimit[39:24], FF_FFFFh}.
- Accesses to addresses that map to both DRAM, as specified by F1x[44,40], 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 the link are routed based on [The DRAM Base/Limit Registers] F1x[44,40], only.
- The appropriate RE or WE bit(s) must be set.
- See also section 2.6.3.1.1 [DRAM and MMIO Memory Space].

Hoisting. When memory hoisting is enabled (F1xF0[DramHoleValid]=1), F1x[44,40][DramLimit] should be set up to account for the memory hoisted above the hole. I.e., F1x[44,40][DramLimit] should be set to F1x[44,40][DramBase] plus the size of the amount of memory plus the hole size (4G minus F1xF0[DramHoleBase]). See section 2.8.8 [Memory Hoisting] for more information about memory hoisting.

F1x40 DRAM Base Address Register

Bits	Description
31:16	**DramBase[39:24]: DRAM base address register bits[39:24]**. Read-only. Reset: 0000h.
15:2	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.

F1x44 DRAM Limit Address Register

Bits	Description
31:16	**DramLimit[39:24]: DRAM limit address register bits[39:24]**. Read-write. Reset: FFFFh.
15:0	Reserved.

F1x[BC:80] Memory Mapped IO Base/Limit Registers

These registers specify up to 8 MMIO address ranges. Each address range is specified by a base/limit register pair. The first set is F1x80 and F1x84, the second set is F1x88 and F1x8C, and so forth. Transaction addresses that are within the specified base/limit range are routed to the IO link. See also section [The Northbridge Routing] 2.6.3.

MMIO mapping rules:

- Transaction addresses are within the defined range if:
  {00h, MMIOBase[39:16], 0000h} <= address[39:0] <= {00h, MMIOLimit[39:16], FFFFh}.
- MMIO regions must not overlap each other.
- Accesses to addresses that map to both DRAM, as specified by F1x[44,40], 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 the link are routed based on [The Memory Mapped IO Base/Limit Registers] F1x[BC:80], only.
- MMIO configuration accesses are not affected by MMIO ranges. See MSRC001_0058.
- 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 2.6.3.1.1 [DRAM and MMIO Memory Space].

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 link specification summary of deadlock scenarios for more information. 0=CPU writes to this MMIO range use the posted channel. This bit does not affect requests that come from the link (the virtual channel of the request is specified by the link request). 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.
6:0	Reserved.

F1x[C4,C0] IO-Space Base/Limit Registers

These registers specify positive decode for IO addresses to the link for transactions resulting from x86-defined IN and OUT instructions. The IO address range is specified by 1 set of base/limit registers. See also section [The Northbridge Routing] 2.6.3.

IO mapping rules:

- IO-space transaction addresses are within the defined range if:
  {IOBase[24:12], 000h} <= address <= {IOLimit[24:12], FFFh} and as specified by the IE bit; or
  if the address is in the range specified by the VE bits.
- The appropriate RE or WE bit(s) must be set.
- See also section 2.6.3.1.2 [IO Space].

F1xC0 IO-Space Base Address Register

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. 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.

F1xC4 IO-Space Limit Address Register

Bits	Description
31:25	Reserved.
24:12	**IOLimit[24:12]: IO limit address register bits[24:12]**. Read-write. Reset: X.
11:0	Reserved.

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]]. DramHoleBase must be > 0. See section 2.8.8 [Memory Hoisting] 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 2.8.8 [Memory Hoisting] for additional programming information.
6:1	Reserved.
0	**DramHoleValid**. Read-write. 1=Memory hoisting is enabled. 0=Memory hoisting is not enabled. See section 2.8.8 [Memory Hoisting] for additional programming information.

F1xF4 VGA Enable Register

Reset: 0000 0000h. All these bits are read-write unless Lock is set.

Bits	Description
31:4	Reserved. [14]:
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. 0=Transactions targeting the VGA-compatible address space are not affected by the state of 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). • An MMIO range (F1x[BC:80]) must not overlap the VGA-compatible address space when F1xF4[VE]=1. • When this bit is set, the state of F1x[C4,C0][VE] is ignored.

F1x1[84:80] NB MMIO Configuration Base Address Register

These addresses form a duplicated access space for MSRC001_0058. See MSRC001_0058[31:0] for F1x180 and MSRC001_0058[63:32] F1x184.

3.5 Function 2 DRAM Controller Registers

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

F2x00 Device/Vendor ID Register

Reset: 1302 1022h.

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

F2x04 Status/Command Register

Reset: 0000 0000h. Read-only.

Bits	Description
31:16	**Status.**
15:0	**Command.**

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.

F2x34 Capabilities Pointer Register

Reset: 0000 0000h.

Bits	Description
31:8	Reserved.
7:0	**CapPtr: capabilities pointer.** Read-only, 00h.

F2x[1,0][4C:40] DRAM CS Base Address Registers

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

These registers along with [The DRAM CS Mask Registers] F2x[1,0][64:60], translate DRAM request addresses (to a DRAM controller) into DRAM chip selects. Supported DIMM sizes are specified in [The DRAM Bank Address Mapping Register] F2x[1,0]80. For more information on the DRAM controllers, see section 2.8 [DRAM Controller (DCT)].

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 24: Logical DIMM, Chip Select, CKE, ODT, and Register Mapping

Base Address Registers	Mask Register	Logical DIMM Number	Chip Select	M[B, A]_ CKE[x]	ODT
F2x[1, 0]40	F2x[1, 0]60	0	MA0_CS_L[0] MB0_CS_L[0]	0	MA0_ODT[0] MB0_ODT[0]
F2x[1, 0]44			MA0_CS_L[1] MB0_CS_L[1]	1	MA0_ODT[1] MB0_ODT[1]
F2x[1, 0]48	F2x[1, 0]64	1	MA1_CS_L[0] MB1_CS_L[0]	0	MA1_ODT[0] MB1_ODT[0]
F2x[1, 0]4C			MA1_CS_L[1]	1	MA1_ODT[1]

The DRAM controller operates on the normalized physical address of the DRAM request. The normalized physical address includes all of the address bits that are supported by a DRAM controller. See also section 2.6.1 [Northbridge (NB) Architecture].

Each base address register specifies the starting normalized address of the block of memory associated with the chip select. Each mask register specifies the additional address bits that are consumed by the block of memory associated with the chip selects. If both chip selects of a 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 2.8.7.1 [Chip Select Interleaving] for more information. The hardware supports the use of lower-order address bits to interleave chip selects if (1) each chip select of the memory system spans the same amount of memory and (2) the number of chip selects of the memory system is a power of two.

System BIOS is required to assign the largest DIMM chip-select range to the lowest normalized address of the DRAM controller. As addresses increase, the chip-select size is required to remain constant or decrease. This is necessary to keep DIMM chip-select banks on aligned address boundaries, regardless as to the amount of address space covered by each chip select.

For each normalized address for requests that enters a DRAM controller, a ChipSelect[i] is asserted if:

```
CSEnable[i] &
     ( {(InputAddr[36:27]   & ~AddrMask[i][36:27]),
        (InputAddr[21:13]   & ~AddrMask[i][21:13])} ==
       {(BaseAddr[36:27]    & ~AddrMask[i][36:27]),
        (BaseAddr[21:13]    & ~AddrMask[i][21:13])} );
```

Bits	Description
31:29	Reserved.
28:19	**BaseAddr[36:27]: normalized physical base address bits [36:27].** Read-write.
18:14	Reserved.
13:5	**BaseAddr[21:13]: normalized physical base address bits [21:13].** Read-write.
4:3	Reserved.
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.
1	Reserved.
0	**CSEnable: chip select enable**. Read-write.

F2x[1,0][64:60] DRAM CS Mask Registers

Reset: 0000 0000h.

See section 2.8.1 [DCT Configuration Registers] for general programming information about DCT configuration registers. See F2x[1,0][4C:40] for information about this register.

Bits	Description
31:29	Reserved.
28:19	**AddrMask[36:27]: normalized physical address mask bits [36:27]**. 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 0036h. See section 2.8.1 [DCT Configuration Registers] for general programming information about DCT configuration registers.

Bits	Description
31:19	Reserved.
18	**DqsRcvEnTrain: DQS receiver enable training mode.** Read-write. Reset: 0. 1=Enable DQS receiver enable training mode. 0 = Normal DQS receiver enable operation.

17	**DllTempAdjTime: DLL Temperature Adjust Cycle Time**. Read-write. Reset: 0. BIOS: 0. This bit selects the DLL temperature adjust cycle time. 0 = 5 ms 1 = 1 ms
16:7	Reserved.
6:4	**RdPadRcvFifoDly: Read Delay from Pad Receive FIFO**. Read-write. Reset: 011b. This field specifies the delay from the DQS receiver enable to the first data being read from the pad receive FIFO. Bits — Read Delay 000b — 0.5 Memory Clocks 001b — 1 Memory Clock 010b — 1.5 Memory Clocks 011b — 2 Memory Clocks 100b — 2.5 Memory Clocks 101b — 3 Memory Clock 110b — 3.5 Memory Clocks 111b — 4 Memory Clocks
3:0	**RdPtrInit: read pointer initial value**. Read-write. Reset: 6h. 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 0h-3h — Reserved 4h — 2 MEMCLKs 5h — 1.5 MEMCLKs 6h — 1 MEMCLK 7h-Fh — Reserved

F2x[1,0]7C DRAM Initialization Register

Reset: 0000 0000h.

BIOS can directly control the DRAM initialization sequence using this register. To do so, BIOS sets EnDramInit to start DRAM initialization. BIOS should then complete the initialization sequence specified in the appropriate JEDEC specification. After completing the sequence, BIOS clears EnDramInit to complete DRAM initialization. BIOS should not assert LDTSTOP_L while EnDramInit is set. Note: setting more than one of the command bits in this register--SendMrsCmd, SendAutoRefresh, and SendPchgAll--at a time results in undefined behavior. See section 2.8.6 [DCT/DRAM Initialization].

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 32 Memory Clocks and then wait until F2x[1,0]98[DctAccessDone] = 1 before programming AssertCke=1. BIOS must clear this bit after DRAM initialization is complete.
30	Reserved.
29	Reserved.
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	Reserved.

26	**SendMrsCmd: send MRS/EMRS command**. Read; Write-1-only; cleared by hardware. 1=The DCT sends the MRS or EMRS commands defined by the MrsAddress and MrsBank fields of this register. This bit is cleared by hardware after the command completes. • This cannot be used for OCD-adjust commands.
25	**SendAutoRefresh: send auto refresh command**. Read; Write-1-only; cleared by hardware. 1=The DCT sends an auto refresh command. This bit is cleared by hardware after the command completes.
24	**SendPchgAll: send precharge all command**. Read; Write-1-only; cleared by hardware. 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. This field is valid only when EnDramInit = 0 and when SendMrsCmd=1; otherwise, MRS/EMRS commands are sent to all chip selects. Bits Definition 000b MRS/EMRS command is sent to CS0 001b MRS/EMRS command is sent to CS1 010b MRS/EMRS command is sent to CS2 011b MRS/EMRS command is sent to CS3 1xxb Reserved.
19	Reserved.
18:16	**MrsBank: bank address for MRS/EMRS commands**. Read-write. This field specifies the data driven on the DRAM bank pins for MRS and EMRS commands.
15:0	**MrsAddress: address for MRS/EMRS commands**. Read-write. This field specifies the data driven on the DRAM address pins 15-0 for MRS and EMRS commands.

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

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

These fields specify DIMM configuration information. Dimm0AddrMap applies to each physical DIMM of logical DIMM 0 (where logical DIMM numbers are specified by [The DRAM CS Base Address Registers] F2x[1,0][4C: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 25 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
31:8	Reserved.
7:4	**Dimm1AddrMap: DIMM 1 address map.** Read-write.
3:0	**Dimm0AddrMap: DIMM 0 address map.** Read-write.

Table 25: DRAM address mapping

		Device size,	Bank			Address																
Bits	CS Size	width	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

Table 25: DRAM address mapping

		Device size,	Bank			Address																
Bits	CS Size	width	2	1	0		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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		Reserved				Row																
						Col																
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		Reserved				Row																
						Col																
0111b	2GB	2Gb, x8	15	14	13	Row	x	17	16	30	29	28	27	26	25	24	23	22	21	20	19	18
		4Gb, x16				Col	x	x	x	x	x	AP	12	11	10	9	8	7	6	5	4	3
1000b		Reserved				Row																
						Col																
1001b		Reserved				Row																
						Col																
1010b	4GB	4Gb, x8	15	14	13	Row	17	16	31	30	29	28	27	26	25	24	23	22	21	20	19	18
						Col	x	x	x	x	x	AP	12	11	10	9	8	7	6	5	4	3
1011b		Reserved				Row																
						Col																

F2x[1,0]88 DRAM Timing Low Register

See section 2.8.1 [DCT Configuration Registers] for general programming information about DCT configuration registers.

Bits	Description
31:24	**MemClkDis: MEMCLK disable**. Read-write. Reset: FFh. 1=Disable the MEMCLK. The bits MemClkDis[7:0] are mapped to package pin names as follows: F2x088 - Channel A MEMCLK control / F2x188 - Channel B MEMCLK control Bit / Package Pin Name / Bit / Package Pin Name 0 Reserved 0 Reserved 1 MA_CLK_H/L[1] 1 MB_CLK_H/L[1] 2 Reserved 2 Reserved 3 Reserved 3 Reserved 4 MA_CLK_H/L[4] 4 MB_CLK_H/L[4] 5 MA_CLK_H/L[5] 5 MB_CLK_H/L[5] 6 Reserved 6 Reserved 7 MA_CLK_H/L[7] 7 MB_CLK_H/L[7] See [The DRAM CS Base Address Registers] F2x[1,0][4C:40] for CKE to DIMM mapping.

23:22	**Trrd: row to row delay (or RAS to RAS delay)**. Read-write. Reset: 00b. This specifies the minimum time between activate commands to different chip-select banks. Bits — Definition 00b — 2 clocks 01b — 3 clocks 10b — 4 clocks 11b — 5 clocks
21:20	**Twr: write recovery time**. Read-write. Reset: 0. This specifies the minimum time from the last data write until the chip-select bank precharge. Bits — Definition 00b — 3 clocks 01b — 4 clocks 10b — 5 clocks 11b — 6 clocks
19:16	**Trc: row cycle time**. Read-write. Reset: 0h. This specifies the minimum time from an activate command to another activate command or an auto-refresh command, all to the same chip-select bank. Bits — Definition 0h — 11 clocks 1h — 12 clocks ... — ... Fh — 26 clocks
15:12	**Tras: row active strobe**. Read-write. Reset: 0h. This specifies the minimum time from an activate command to a precharge command, both to the same chip-select bank. Bits — Definition 0h — Reserved 1h — Reserved 2h — 5 clocks 3h — 6 clocks ... — ... Fh — 18 clocks
11:10	**Trtp: read to precharge time**. Read-write. Reset: 00b. Read CAS to Precharge. It measures 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. • BIOS: If (F2x[1,0]94[MemClkFreq]<=001b) then 00b, else 10b. Bits — Definition 0xb — 2 clocks for burst length of 32 bytes 4 clocks for burst length of 64 bytes 1xb — 3 clocks for burst length of 32 bytes 5 clocks for burst length of 64 bytes
9:7	**Trp: row precharge time**. Read-write. Reset: 000b. This specifies the minimum time from a precharge command to an activate command or auto-refresh command, both to the same bank. Bits — Definition 00xb — 3 clocks 01xb — 4 clocks 10xb — 5 clocks 11xb — 6 clocks

6:4	**Trcd: RAS to CAS delay**. Read-write. Reset: 000b. This specifies the time from an activate command to a read/write command, both to the same bank. Bits Definition x00b 3 clocks x01b 4 clocks x10b 5 clocks x11b 6 clocks
3:0	**Tcl: CAS latency**. Read-write. Reset: 0h. This specifies the time from the CAS assertion for a read cycle until data return (from the perspective of the DRAM devices). Write CAS latency is always read CAS latency minus 1. Bits Definition 0000b Reserved 0001b Reserved 0010b 3 clocks 0011b 4 clocks 0100b 5 clocks 0101b 6 clocks 0110b - 1111b Reserved

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

Reset: 0000 000Ah.

See section 2.8.2 [DCT Configuration Registers] on page 14 for general programming information about DCT configuration registers.

Bits	Description
31:26	Reserved
25:23	**Trfc1: auto-refresh row cycle time for logical DIMM 1**. Read-write. See Trfc0.
22:20	**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][4C:40]. The recommended programming of this register varies based on DRAM density and speed. Bits Definition 000b 75 ns (all speeds, 256 Mbit) 001b 105 ns (all speeds, 512 Mbit) 010b 127.5 ns (all speeds, 1 Gbit) 011b 195 ns (all speeds, 2 Gbit) 100b 327.5 ns (all speeds, 4 Gbit) 101b-111b Reserved
19	Reserved.
18	**DisAutoRefresh: disable automatic refresh**. Read-write. 1=Automatic refresh is disabled. This is sometimes useful during electrical characterization.
17:16	**Tref: refresh rate.** Read-write. This specifies the average time between refresh requests to all DRAM devices. Bits Definition 00b Undefined behavior. 01b Reserved 10b Every 7.8 us. 11b Every 3.9 us.

15:14	**Trdrd[1:0]: read-to-read timing**. Read-write. BIOS: 00b. 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. Bits Definition 00b 2 clocks (1 idle cycle on the data bus) 01b 3 clocks (2 idle cycles on the data bus) 10b 4 clocks (3 idle cycles on the data bus) 11b 5 clocks (4 idle cycles on the data bus)
13:12	**Twrwr[1:0]: write-to-write timing**. Read-write. BIOS: 01b. 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. Bits Definition 00b 1 clock (no idle cycles on the data bus) 01b 2 clocks (1 idle cycle on the data bus) 10b 3 clocks (2 idle cycles on the data bus) 11b 4 clocks (3 idle cycles on the data bus)
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. 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. See section [The Twrrd (Write-to-Read DIMM Termination Turn-around)] 2.8.6.2.1 for information on how to program this field. Bits Definition 00b 1 clock (1 idle cycle on the data bus) 01b 2 clocks (2 idle data bus cycles) 10b 3 clocks (3 idle data bus cycles) 11b 4 clocks (4 idle data bus cycles)
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 the last non-masked data strobe of the write to the rising clock edge of the next read command. Bits Definition 00b Reserved 01b 1 clocks 10b 2 clocks 11b 3 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. See section [The TrwtTO (Read-to-Write Turnaround for Data, DQS Contention)] 2.8.6.2.2 for information on how to program this field. Bits Definition 0000b Reserved 0001b 3 clocks 0010b 4 clocks ... ... 0111b 9 clocks 1000b - 1111b Reserved
3:0	**TrwtWB: read-to-write turnaround for opportunistic write bursting**. Read-write. BIOS: Fh. Specifies the minimum number of cycles from the last read operation seen by the DCT scheduler to the following write operation. The purpose of this field is to hold off write operations until several cycles have elapsed without a read cycle; this may result in a performance benefit. In absence of opportunistic write bursting the DCT waits until F2x11C[DctWrLimit] has been reached before scheduling writes to DRAM. Bits Definition 0000b Reserved. 0001b 1 clock ... ... 1010b 10 clocks ... ... 1111b 15 clocks

F2x[1,0]90 DRAM Configuration Low Register

Reset: 0000 0000h. See section 2.8.1 [DCT Configuration Registers] 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	**IdleCycInit: idle cycle counter initial value**. Read-write. BIOS: 11b. This specifies the initial number of MEMCLK cycles during which an open page of DRAM is not accessed before it may be closed by the dynamic page close logic. This field is ignored if F2x[1,0]90[DynPageCloseEn] = 0. Bits Definition 00b 16 clocks 01b 32 clocks 10b 64 clocks 11b 96 clocks • F2x090[IdleCycInit] and F2x190[IdleCycInit] must always be set to the same value. See 2.8.1 [DCT Configuration Registers].

20	**DynPageCloseEn: dynamic page close enable**. Read-write. BIOS: 0. 1=The DRAM controller dynamically determines when to close open pages based on the history of that particular page and F2x[1,0]90[IdleCycInit]. 0=Any open pages not auto-precharged by the DRAM controller are automatically closed after 128 clocks of inactivity. • F2x090[DynPageCloseEn] and F2x190[DynPageCloseEn] must always be set to the same value. See 2.8.1 [DCT Configuration Registers].
19:11	Reserved.
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. • BurstLength32 must be 0 when the DRAM interface is 128 bits wide; this bit interacts with F2x110[DctGangEn] as follows: BurstLength32 DctGangEn 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 Reserved.
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.
8	Reserved.
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.
6	Reserved.
5:4	**DramTerm: DRAM termination**. Read-write. This specifies the programming of the DRAM termination value (Rtt) when the EMRS command is issued during DRAM initialization (F2x[1,0]90[InitDram]). Bits Definition 00b On die termination disabled. 01b 75 ohms. 10b 150 ohms. 11b 50 ohms.
3:2	Reserved.
1	**ExitSelfRef: exit self refresh (after suspend to RAM) 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. 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 2.8.6.3 [DRAM Device Initialization] for more details.

F2x[1,0]94 DRAM Configuration High Register

Reset: 0F00 0000h.

See section 2.8.1 [DCT Configuration Registers] for general programming information about DCT configuration registers.

Bits	Description
31:28	**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 specifications. The field encoding is as follows: Bits — Window size 0000b — No tFAW window restriction. 0001b — 8 MEMCLK cycles. 0010b — 9 MEMCLK cycles. ... — ... 1101b — 20 MEMCLK cycles. 1110b - 1111b — Reserved See section 2.8.6.2.3 [FourActWindow (Four Bank Activate Window or tFAW)] for information on how to program this field.
27:24	**DcqBypassMax: DRAM controller queue bypass maximum**. Read-write. Reset: Fh. BIOS: Eh. The DRAM controller arbiter normally allows transactions to pass other transactions in order to optimize DRAM bandwidth. This field specifies the maximum number of times that the oldest memory-access request in the DRAM controller queue may be bypassed before the arbiter decision is overridden and the oldest memory-access request is serviced instead. If (DcqBypassMax>0), then the oldest request may be bypassed no more than ((DcqBypassMax+1)*4) times. If (DcqBypassMax=0), then the oldest request is never bypassed. • F2x094[DcqBypassMax] and F2x194[DcqBypassMax] must always be set to the same value. See 2.8.1 [DCT Configuration Registers]. Bits — Definition 0h — No bypass; the oldest request is never bypassed. 1h — The oldest request may be bypassed no more than 8 times. 2h — The oldest request may be bypassed no more than 12 times. ... — ... Eh — The oldest request may be bypassed no more than 60 times. Fh — The oldest request may be bypassed no more than 64 times.
23	**ProcOdtDis: processor on-die termination disable**. Read-write. 1=The processor-side on-die termination is disabled. 0=Processor-side on-die termination enabled. See F2x[1,0]9C_x00[ProcOdt] for ODT definitions.

22	**BankSwizzleMode: bank swizzle mode**. Read-write. BIOS: 1. 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 25 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. **64-bit unganged mode example** For example, encoding 02h of Table 25 for 64-bit unganged mode 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}. If, for example, [18:13]=110001b, then Bank[1:0] = {0 ^ 0 ^ 1, 1 ^ 0 ^ 1} = {1, 0}. **128-bit ganged mode example** For example, encoding 02h of Table 25 for 128-bit unganged mode would be remapped from bank[1:0]={A15, A14} to the following for a 64-bit DCT: Bank[1:0] = {A15 ^ A17 ^ A19, A14 ^ A16 ^ A18}. If, for example, [19:14]=110001b, then Bank[1:0] = {0 ^ 0 ^ 1, 1 ^ 0 ^ 1} = {1, 0}.
21	Reserved.
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	**CkeOdtMapMode: CKE to ODT mapping mode**. Read-write. When set, this bit is used to specify ODT to chip select and CKE associations to enhance power savings for specific DIMM topologies. 0=Normal chip select, clock enable, and ODT associativity. See section [The DRAM CS Base Address Registers] F2x[1,0][4C:40]. 1=ODT mapping is reconfigured based on the following: • For a single channel with two DIMMs on DCT0: ODT0 is associated with CS0, CS1, and CKE0 for DIMM0. ODT1 is associated with CS2, CS3, and CKE1 for DIMM1. • For 1 DIMM and 1 Rank soldered down: ODT0 is associated with CS0, CS1, and CKE0 for DIMM0. ODT1 is associated with CS2 and CKE1 for soldered down rank. Note: the second ODT input on the DIMM should be connected to VSS. See [The DRAM CS Base Address Registers] F2x[1,0][4C:40] for CKE, ODT, and CS to DIMM mapping.
18:17	Reserved.

16	**PowerDownMode: power down mode**. Read-write. BIOS: 1. 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. 1b Chip select CKE control mode. A chip select or pair of chip selects is placed in power down mode when no transactions are pending for the chip select(s). This mode is expected to be used in mobile systems: - CKE0 is associated with CS0 in 2-rank systems. - CKE1 is associated with CS1 in 2-rank systems. - CKE0 is associated with CS0 and CS2 in 3 and 4-rank systems. - CKE1 is associated with CS1 and CS3 in 3 and 4-rank systems. • Note: see section [The DRAM CS Base Address Registers] F2x[1,0][4C:40] for more information on DIMM system configurations and power down mode behavior.
15	**PowerDownEn: power down mode enable**. Read-write. BIOS: 1. 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.
13:8	Reserved.
7:4	**MaxAsyncLat: maximum asynchronous latency**. Read-write. This field should be programmed by system BIOS to specify the maximum roundtrip 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 core clock domain. See section 2.8.6.4.3 [Maximum Asynchronous Latency (MaxAsyncLat)] for information on how to program this field. Bits Maximum Latency 0000b 0 ns 0001b 1 ns 1111b 15ns
3	**MemClkFreqVal: memory clock frequency valid**. Read-write. System BIOS should set this bit when 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. Note: this bit should not be set if the DCT is disabled. BIOS must change each DCT’s operating frequency in order. See section 2.8.6.3.

2:0	**MemClkFreq: memory clock frequency**. Read-write. This field specifies the frequency of the DRAM interface (MEMCLK). • The actual DRAM frequency may be less than the programmed frequency as a function of the main PLL frequency; see Table 19 on page 70. • If F2x194[MemClkFreqVal]=1 in unganged mode, then both F2x094[MemClkFreq] and F2x194[MemClkFreq] must be initialized to the same value. If F2x194[MemClkFreqVal]=1 in ganged mode or if F2x194[MemClkFreqVal]=0, then F2x194[MemClkFreq] is unused. Bits Definition 000b Reserved. 001b 266 MHz 010b 333 MHz 011b 400 MHz 100b-111b 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[1B: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[1B: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.

See section 2.8.1 [DCT Configuration Registers] for general programming information about DCT configuration registers. Note, however, that F2x198, F2x098, F2x19C_x[1B:00], and F2x09C_x[1B: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 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 registers. 1=Write one of the F2x[1, 0]9C_x 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: 0011 1222h. See F2x[1,0]98 for register access information.

Bits	Description
31:30	Reserved.
29:28	**ProcOdt: processor on-die termination**. Read-write. Reset: 00b. This field specifies the resistance of the on-die termination resistors. This field is valid only when F2x[1,0]94[ProcOdtDis]=0. Bits Definition 00b 300 ohms +/- 20% 01b 150 ohms +/- 20% 10b 75 ohms +/- 20% 11b Reserved
27:22	Reserved.
21:20	**DqsDrvStren: DQS drive strength**. Read-write. Reset: 01b. This field specifies the drive strength of the DQS pins. 00b 0.75x 10b 1.25x 01b 1.0x 11b 1.5x
19:18	Reserved.
17:16	**DataDrvStren: data drive strength**. Read-write. Reset: 01b. This field specifies the drive strength of the DRAM data pins. 00b 0.75x. 10b 1.25x 01b 1.0x 11b 1.5x
15:14	Reserved.
13:12	**ClkDrvStren: MEMCLK drive strength**. Read-write. Reset: 01b. This field specifies the drive strength of the MEMCLK pins. 00b 0.75x. 10b 1.25x 01b 1.0x 11b 1.5x
11:10	Reserved.
9:8	**AddrCmdDrvStren: address/command drive strength**. Read-write. Reset: 10b. 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. Reset: 10b. 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. Reset: 10b. 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[02:01] DRAM Write Data Timing [High:Low] Registers

Reset: 1717 1717h. See F2x[1,0]98 for register access information.

These registers control the timing of write data with respect to DQS. See section 2.8.6.4 [DRAM Training] for information on how to use these registers. Timing for all DIMMs on each channel is controlled by these registers. F2[1, 0]9C_x01 are the DRAM Write Data Timing Low Registers and control bytes[3:0] of data. F2[1, 0]9C_x02 are the DRAM Write Data Timing High Registers and they control bytes[7:4] of data. Each of the fields in these registers are encoded as follows:

Write Data Delay Timing (WrDatDlyByte):

00_0000b	No delay
00_0001b	1/96 MEMCLK delay
00_0010b	2/96 MEMCLK delay
...	...
10_1111b	47/96 MEMCLK delay
11_0000b-11_1111b	Reserved

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

F2x[1,0]9C_x04 DRAM Address/Command Timing Control Register

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.

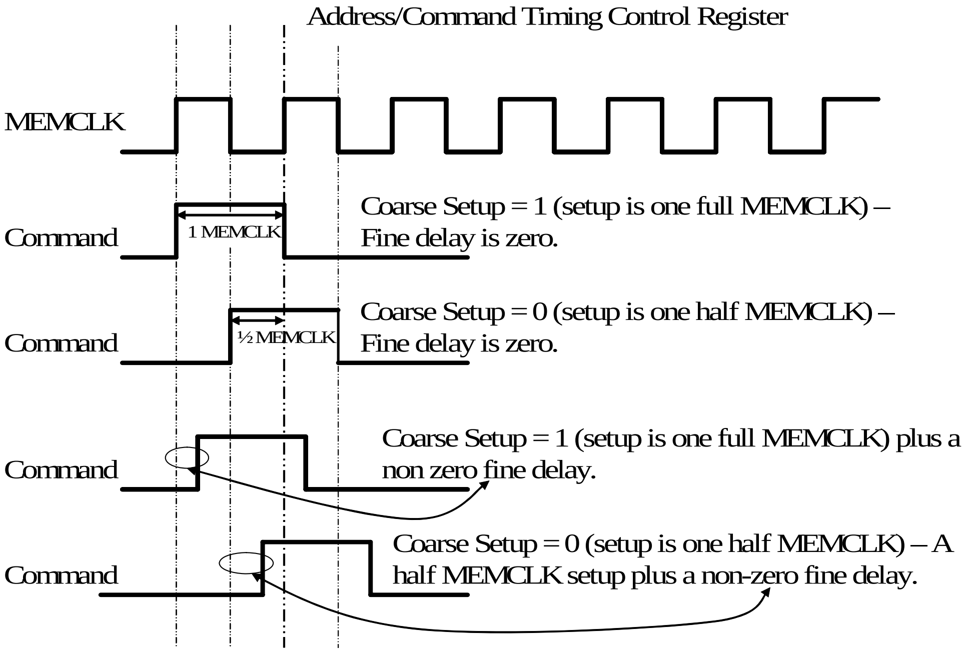
Address/Command Timing Control Register
MEMCLK
Command
1 MEMCLK
Coarse Setup = 1 (setup is one full MEMCLK) – Fine delay is zero.
Command
½ MEMCLK
Coarse Setup = 0 (setup is one half MEMCLK) – Fine delay is zero.
Command
Coarse Setup = 1 (setup is one full MEMCLK) plus a non zero fine delay.
Command
Coarse Setup = 0 (setup is one half MEMCLK) – A half MEMCLK setup plus a non-zero fine delay.

Figure 10: Address/Command Timing at the Processor Pins

2T timing is controlled by F2x[1,0]94[SlowAccessMode].

Bits	Description
31:29	Reserved.
28	**AbyteDllMaxPhases: 32/64 phase mode select**. Read-write. BIOS: 0. 0= The 8 Abyte DLLs use 32 max phases per unit interval. 1= The 8 Abyte DLLs use 64 max phases per unit interval.
27:22	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[06:05] DRAM Read DQS Timing Control [High:Low] Registers

Reset: 1717 1717h. See F2x[1,0]98 for register access information. These registers control the timing of read (input) DQS signals with respect to data per channel. See section 2.8.6.4 [DRAM Training] for information on how to use these registers. Timing for all DIMMs on each channel is controlled by these registers. F2[1, 0]9C_x05 are the DRAM Read DQS Timing Control Low Registers; they control DQS for bytes[3:0] of data. F2[1, 0]9C_x06 are the DRAM Read DQS Timing Control High Registers; they control DQS for bytes[7:4] of data. Each of the fields in these registers specify how much DQS is delayed with respect to data as follows:

00_0000b	No delay
00_0001b	1/96 MEMCLK delay
00_0010b	2/96 MEMCLK delay
...	...
10_1111b	47/96 MEMCLK delay
11_0000b-11_1111b	Reserved

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[2B:10] DRAM DQS Receiver Enable Timing Control Registers

Reset: 0000 0000h. See F2x[1,0]98 for register access information.

These registers are organized as two groups of registers, one groups for each DIMM. DIMM numbers are specified by [The DRAM CS Base Address Registers] F2x[1,0][4C:40]. The value applied to byte 0 of each register is applied to all byte lanes of the channel.

DctOffset	Register
0000_0010h	DRAM DQS Receiver Enable Timing Control Low DIMM 0, DCT [1,0]: (bytes 0-7)
0000_00[12:11]h	Reserved
0000_0013h	DRAM DQS Receiver Enable Timing Control Low DIMM 1, DCT [1,0]: (bytes 0-7)
0000_00[2B:14]h	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 2.8.6.4 [DRAM Training] for information on how to use these registers. The field delay value is applied to all byte lanes. 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:

Gross timing (for DqsRcvEnGrossDelay):
Delay = DqsRcvEnGrossDelay * 0.5MEMCLKs, ranging from 0 to 3.5 MEMCLKs

Fine timing (for DqsRcvEnFineDelay):
Delay = DqsRcvEnFineDelay * 1/64 MEMCLKs, ranging from 0 to 31/64 MEMCLKs

Bits	Description
31:8	Reserved.
7:5	**DqsRcvEnGrossDelay: DQS receiver enable gross delay**. Read-write.
4:0	**DqsRcvEnFineDelay: DQS receiver enable fine delay**. Read-write.

F2x[1,0]A0 DRAM Controller Miscellaneous Register

Reset: 0000 0001h. See section 2.8.1 [DCT Configuration Registers] for general programming information about DCT configuration registers.

Bits	Description
31:8	Reserved.

7:6	**DllPwrDwn: DLL power down**. Read-write. BIOS: 10b. According to the following table, a DRAM channel indicates that the DLL can be powered down. The processor powers down the DLL's for all enabled DRAM channels when all enabled DRAM channels indicate that the DLL can be powered down. 00b Disabled. 01b Reserved. 10b Enable if in self refresh mode. 11b Enable if in self refresh mode and F3x[84:80][DramDllVRegPwrDwn]=1 for the ACPI state. • The DLL is powered down when the DLL voltage regulator is powered down, as controlled by DllVRegPwrDwn. • If channel A is enabled then F2x1A0[DllPwrDwn] is Reserved and F2x0A0[DllPwrDwn] controls both channels. If channel A is disabled then F2x1A0[DllPwrDwn] controls channel B.
5:4	**DllVRegPwrDwn: DLL voltage regulator power down**. Read-write. BIOS: 10b. According to the following table, a DRAM channel indicates that the DLL voltage regulator can be powered down. The processor powers down the DLL voltage regulator when all enabled DRAM channels indicate that the DLL voltage regulator can be powered down. 00b Disabled. 01b Enable if in self refresh mode. 10b Enable if in self refresh mode and F3x[84:80][DramDllVRegPwrDwn]=1 for the ACPI state. 11b Reserved. • If channel A is enabled then F2x1A0[DllVRegPwrDwn] is Reserved and F2x0A0[DllVRegPwrDwn] controls both channels. If channel A is disabled then F2x1A0[DllVRegPwrDwn] controls channel B.
3	**PhyClkDivInSR: phy clock divider in self refresh**. Read-write. BIOS: 1. 1= DDR PHY clock changed to be Main PLL Clock/512 and NCLK changed to Main PLL Clock/64 when in self refresh mode. 0=No change to prenclk2x or NCLK in self refresh mode. • F2x0A0[PhyClkDivInSR] and F2x1A0[PhyClkDivInSR] must always be set to the same value, even if one of the DCT channels is disabled. See 2.8.1 [DCT Configuration Registers].
2:0	Reserved.

F2xA4 DRAM Controller Temperature Throttle Register

Reset: 0000 0000h.

Bits	Description
31:3	Reserved.
2:1	**ThrottleEn[1:0]:DRAM throttle enable**. Read-write. Reset: 00b. When the MEMHOT_L pin is asserted throttle both DCT channels to the average utilization specified by ThrottleEn[1:0]. Throttling is accomplished by reducing command (Precharge, Activate, Read(Ap), Write(Ap)) issue bandwidth. See "DRAM Thermal Protection (MEMHOT_L)" on page 69. 00b: 100% (No throttling) 01b: 50% (New commands can be issued no more frequently than 1 every 2 MEMCLKs.) 10b: 25% (New commands can be issued no more frequently than 1 every 4 MEMCLKs.) 11b: 12% (New commands can be issued no more frequently than 1 every 8 MEMCLKs.)
0	**DoubleTrefRateEn: double Tref rate enable**. Read-write. Reset: 0. 1=Tref forced to 3.9us auto-refresh interval when the MEMHOT_L pin is asserted. See "DRAM Thermal Protection (MEMHOT_L)" on page 69.

F2x10C Interleaved Region Base/Limit Register

Reset: 0000 0000h. Enables swapping a region below 4G with the same sized region located at the bottom of memory.

- The size of the swapped high region must be a integer multiple of 128M, defined to be {F2x10C[IntLvRegionBase],000b,000000h} to {F2x10C[IntLvRegionLimit],111b,FFFFFFh}.
- The size and location of the low region is defined to be 0000_0000h to {(F2x10C[IntLvRegionLimit]-F2x10C[IntLvRegionBase],111b,FFFFFFh}.
- The swapped high region can not overlap above MSRC001_001A[TOM].
- The swapped region must be all DRAM. I.e. No IO hole.
- Interleaving must be enabled and the DCTs must be of unequal size.
- See F2x110[DctSelIntLvEn]. See section 2.8.8 [Memory Hoisting] for programming information.

Bits	Description
31:16	Reserved.
15:11	**IntLvRegionLimit: interleaved region limit address**. Read-write. Interleave region limit address [31:27].
10:8	Reserved.
7:3	**IntLvRegionBase: interleaved region base address**. Read-write. Interleave region base address [31:27].
2:1	Reserved.
0	**IntLvRegionEn: interleaved region remap enable**. Read-write. 1=Enable remapping into the DCT-interleaved region.

F2x110 DRAM Controller Select Low Register

Reset: 0000 0000h.

Bits	Description
31:24	Reserved.
23:11	**DctSelBaseAddr[39:27]: DRAM controller select base address bits[39:27]**. Read-write. If the DCTs are unganged (DctGangEn=0), this delineates the address range of the two DCTs by specifying the base address of the upper address range. See section 2.8.8 [Memory Hoisting] for additional programming information.
10	**MemCleared: memory cleared**. Read-only. 1=Memory has been cleared since the last warm reset. This bit is set by MemClrInit. See MemClrInit below.
9	**MemClrBusy: memory clear busy**. Read-only. 1=Memory clear operation in either of the DCTs is in progress. Reads or writes to DRAM while the memory clear operation is in progress result in undefined behavior.
8	**DramEnable: DRAM enabled**. Read-only. 1=All of the used DCTs are initialized (see section 2.8.6.3 [DRAM Device Initialization]) 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. BIOS: 10b. 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. 01b Address bit 12. 10b Hash: exclusive OR of address bits[20:16, 6]. 11b Reserved.
5	Reserved.
4	**DctGangEn: DRAM controller ganging enable**. Read-write. BIOS: 0. 1=Ganged. Both DCTs are ganged to form a single double-width DDR interface. 0=Unganged. The DCTs operate independently. This also affects how DCT configuration registers; see section 2.8.1 [DCT Configuration Registers]. Note, if ganging is to be enabled, this bit must be set prior to accessing any DCT registers. • Unganged is required if F2x[1,0]90[BurstLength32]=1.
3	**MemClrInit: memory clear initialization**. Write-only; reads as 0. 1=The processor writes 0's to all locations of system memory attached to the processor as follows and sets the MemCleared bit: • If F1xF0[DramHoleValid]=0 then memory is cleared: • from 0 ({F1x[44,40][DramBase[39:24]], 00_0000h}) to {F1x[44,40][DramLimit[39:24]], FF_FFFFh}. • If F1xF0[DramHoleValid]=1 then memory is cleared: • from 0 ({F1x[44,40][DramBase[39:24]], 00_0000h}) to ({00h,F1xF0[DramHoleBase[31:24]],000000h}-1). • from 01_00000000h (4 GB) to {F1x[44,40][DramLimit[39:24]], FF_FFFFh}. • Undefined behavior may result if the DRAM hole is enabled (F1xF0[DramHoleValid]=1) and F1x[44,40][DramLimit[39:32]]=00h. (The DRAM hole is enabled and the DRAM limit is below 4 GB.) • 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. • BIOS must set the following registers before setting MemClrInit: • [The DRAM Base/Limit Registers] F1x[44,40] • [The DRAM Hole Address Register] F1xF0 • [The DRAM Bank Address Mapping Register] F2x[1,0]80 • [The DRAM CS Base Address Registers] F2x[1,0][4C:40] • [The DRAM CS Mask Registers] F2x[1,0][64:60] • [The DRAM Controller Select Low Register] F2x110 • [The DRAM Controller Select High Register] F2x114 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 (specified by [The DRAM Base/Limit Registers] F1x[44,40]) to DctSelBaseAddr (if enabled). • BIOS: (~F2x094[DisDramInterface] && ~F2x194[DisDramInterface] && ~F2x10C[DctGangEn]). • If the amount of memory connected to each of the DCTs is different, then channel interleaving may be supported across the address range that includes both DCTs, the top of which is specified by DctSelBaseAddr; the remainder of the address space, above DctSelBaseAddr, would then be allocated to only the DCT connected to the larger amount of memory, specified by DctSelHi. • The previously mentioned non-interleaved remainder of the memory is normally at the top end of physical memory. For performance reasons, related to UMA memory being mapped onto a non-interleaved memory region, a mechanism is provided to swap a region of memory. See F2x10C [Interleaved Region Base/Limit Register].
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). The high range DCT is called DctHi, the low range DCT is called DctLo. 0=High addresses go to DCT0. 1=High addresses go to DCT1.
0	**DctSelHiRngEn: DRAM controller select high range enable**. Read-write. 1=Enables addresses greater than or equal to DctSelBaseAddr[39:27] to be used to select between DCT0 and DCT1; DctSelHi specifies which DCT occupies the high range. Note: if DctGangEn=1, then this bit is not used.

F2x114 DRAM Controller Select High Register

Reset: 0000 0000h.

Bits	Description
31:24	Reserved.
23:10	**DctSelBaseOffset[39:26]: DRAM controller select base offset address bits[39:26]**. 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 2.8.8 [Memory Hoisting] 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 be able to meet the isochronous bandwidth and latency requirements. The variable priority allows the memory controller to optimize DRAM transactions until isochronous traffic reaches a time threshold and must be serviced more quickly.

If a write requires a read-modify-write, arbitration occurs separately for the read and the write and the read has the same priority level as the write. If the priority of the write is changed for a read-modify-write then the priority of the read is changed as well to maintain the same priority was the write.

Bits	Description
31:28	**MctVarPriCntLmt: variable priority time limit**. Read-write. Reset: 0h. BIOS: 1h. 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
27:18	Reserved.
17	**EnSameAddrSerDROp: enable same address serialization for display refresh requests**. Read-write. Reset: 0. BIOS: 0. 1=Same address serialization is enabled for DR requests. 0=Same address serialization is disabled for DR requests. Older requests will not be prevented from passing due to a 39:6 address match. DR requests will not be prevented from passing older requests due to a 39:6 address match. See F3x188[EnSameAddrSerBehindDROp].
16	**En64BitMmioRd: disable 64 bit MMIO read**. Read-write. Reset: 0. BIOS: 1. 1=Aligned 64-bit read accesses converted to 1 RdSizedDword transaction of Length= 2 DW's. 0=Aligned 64-bit read accesses converted to 2 RdSizedByte transactions of Length= 4 Bytes. (K8 behavior)
15:14	**MctPriTrace: NB trace-mode request priority**. Read-write. Reset: 10b. BIOS: 10b.
13:12	**MctPriDR: display refresh VC set read priority**. Read-write. Reset: 10b. BIOS: 11b. • See [The Display Refresh And IFCM] 2.6.3.2.1.
11:10	**MctPriWr: default write priority**. Read-write. Reset: 01b. BIOS: 01b.
9:8	**MctPriDefault: default non-write priority**. Read-write. Reset: 00b. BIOS: 00b.
7:6	**MctPriHiWr: high-priority VC set write priority**. Read-write. Reset: 00b. BIOS: 10b. • See [The Display Refresh And IFCM] 2.6.3.2.1.
5:4	**MctPriHiRd: high-priority VC set read priority**. Read-write. Reset: 10b. BIOS: 10b. • See [The Display Refresh And IFCM] 2.6.3.2.1.
3:2	**MctPriCpuWr: CPU write priority**. Read-write. Reset: 01b. BIOS: 01b.
1:0	**MctPriCpuRd: CPU read priority**. Read-write. Reset: 00b. BIOS: 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 specifies how writes may be burst from the DCT to improve DRAM efficiency. Bursting writes improves DRAM efficiency by minimizing the read-to-write turnaround time and the interference that non-latency critical stores have on latency critical loads. When the number of writes in the DCT reaches the value specified in DctWrLimit, then they become eligible for scheduling. Once eligible for scheduling, the priority based reorder algorithm picks the optimal write to increase DRAM bandwidth.

Rules regarding write bursting:

- Write bursting mode only applies to low-priority writes. Medium and high priority writes are not withheld from the DCTs for write bursting.
- If write bursting is enabled, writes stay in the DCT until the threshold specified by DctWrLimit is reached. Once the threshold is reached, all writes in DCT are converted to medium priority.

- Any write in the DCT that matches the address of a subsequent access is promoted to either medium priority or the priority of the subsequent access, whichever is higher.

Memory prefetching. The DRAM prefetcher detects stride patterns in the stream of requests and then, for predictable stride patterns, generates prefetch requests. A stride pattern is a pattern of requests through system memory that are the same number of cache lines apart. The prefetcher supports strides of -4 to +4 cache lines, which can include alternating patterns (e.g. +1, +2, +1, +2), and can prefetch 2 cache lines ahead depending on the confidence. The prefetcher tracks up to 8 stride patterns simultaneously. Each of these stride patterns has a confidence level associated with it that is modified by how many requests match the stride pattern and is used to determine whether to fetch 2 ahead. The prefetcher behaves according to the following rules for each request:

- When a request matches a stride pattern, then:
    - The confidence level is incremented if less than PrefConfSat.
    - One request ahead will be made if the confidence level < PrefConf. Two requests ahead will be made if the confidence level >= PrefConf.
    - Before the prefetch request is issued to the DCT, the prefetch address is checked that it still falls within the same 4 KB page as the request. If it the prefetch address crosses into a different 4 KB page, then the prefetch is squashed and the stride pattern is deallocated.
- Each time a request is received within +/- 4 cache lines of the last requested cache line in the pattern that does not match the pattern, then the confidence level is decreased by one.
- Each request that is not within the same 4 KB page as the last requested cache line of the stride patterns tracked initiates a new stride pattern by displacing one of the existing least-recently-used stride patterns.
    - The confidence level is 0 when allocated.
- Each request that is within the same 4 KB page but outside the -4 to +4 cache line range (including the exact same cache line) of the last requested cache line of all the stride patterns tracked is ignored.

Bits	Description
31	Reserved.
30	**FlushWr: flush writes command**. Read; Write-1-only; cleared by hardware. Reset: 0. BIOS: 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. BIOS: 0. 1=Causes write bursting to be cancelled and all outstanding writes to be flushed to DRAM when in the stop-grant state. This bit should be set to ensure writes are drained to DRAM before reset is asserted for the suspend-to-RAM state.
28:25	Reserved.
24:22	**PrefConf: prefetch two-ahead confidence**. Read-write. Reset: 011b. BIOS: 1h. Confidence level required to issue one prefetch which is 2 strides ahead. Steady state keeps 2 prefetches ahead. 000b = Reserved. 001b = 2 ... 111b = 14
21:20	Reserved.

19:18	**PrefConfSat: prefetch confidence saturation**. Read-write. Reset: 00. BIOS: 0h. Specifies the point at which prefetch confidence level saturates and stops incrementing. 00b = 15 01b = 7 10b = 3 11b = Reserved.
17:14	Reserved.
13	**PrefIoDis: prefetch IO-access disable**. Read-write. Reset: 1. BIOS: 1. 1=Disables IO requests from triggering prefetch requests.
12	**PrefCpuDis: prefetch CPU-access disable**. Read-write. Reset: 1. BIOS: 0. 1=Disables CPU requests from triggering prefetch requests.
11:7	Reserved.
6:2	**DctWrLimit: memory controller write-burst limit**. Read-write. Reset: 18h. BIOS: 19h. Specifies the number of low-priority writes held in the memory controller queue before they are burst into the DCTs. 0xxxxb = Reserved. 10000b = 16. 10001b = 15. ... 11110b = 2. 11111b = Write bursting disabled.
1:0	Reserved.

3.6 Function 3 Miscellaneous Configuration Registers

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

F3x00 Device/Vendor ID Register

Reset: 1303 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 F3xE8[SVM Capable]=1 then F3x04[Status[20]] is 1; otherwise it is 0.
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 F3xE8[SVM Capable]=1 then F3x34[CapPtr] is 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 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.

[The MCA NB Control Register] 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, and [The MCA NB Address Low Register] F3x50 as they occur, as specified by F3x4C[Over]. Uncorrectable errors immediately result in a Machine Check exception.

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 table walk.
26	Reserved.
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.
24:20	Reserved.
19	**RtryHt0En: link 0 retry reporting enable**. Read-write. 1=Enables MCA reporting of retries on link 0.
18	Reserved.

17	**HtDataEn: link data error reporting enable**. Read-write. 1=Enables MCA reporting of packets with data errors detected on the link.
16	**HtProtEn: link protocol error reporting enable**. Read-write. 1=Enables MCA reporting of protocol errors detected on the link. This enable should be cleared before initiating a warm reset to avoid reporting spurious errors due to RESET# signal skew.
15:14	Reserved.
13	**DevErrEn: DEV error reporting enable**. Read-write. 1=Enables MCA reporting of SVM DEV errors.
12	**WDTRptEn: watchdog timer error reporting enable**. Read-write. 1=Enables MCA reporting of watchdog timer errors. The watchdog timer checks for NB system accesses for which a response is expected but no response is received. See [The MCA NB Configuration Register] 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	Reserved.
9	**TgtAbortEn: target abort error reporting enable**. Read-write. 1=Enables MCA reporting of target aborts to a link. The NB returns an error response back to the requestor with any associated data all 1s independent of the state of this bit.
8	**MstrAbortEn: master abort error reporting enable**. Read-write. 1=Enables MCA reporting of master aborts to a link. The NB returns an error response back to the requestor with any associated data all 1s independent of the state of this bit.
7:6	Reserved.
5	**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 link with sync packets after detecting a sync packet on an incoming link independent of the state of this bit.
4:3	Reserved.
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 for ramifications. The NB floods its outgoing link with sync packets after detecting a CRC error on an incoming link independent of the state of this bit.
1:0	Reserved

F3x44 MCA NB Configuration Register

Reset: 0000 0000h.

Bits	Description
31	**NbMcaLogEn: NB MCA log enable**. Read-write. 1=Enables logging (but not reporting) of NB MCA errors even if MCA is not globally enabled.
30	Reserved.

29	**DisMstAbtCpuErrRsp: master abort CPU error response disable**. Read-write. 1=Disables master abort reporting through the MCA error-reporting banks.
27	**NbMcaToMstCpuEn: machine check errors to master CPU only**. Read-write. 1=MCA errors in a CMP device are reported only to core 0, and the NB MCA registers in MSR space (MSR0000_0410, MSR0000_0411, MSR0000_0412, MSR0000_0413, MSRC001_0048) are only accessible from core 0; 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] for a description of MSR space and PCI-defined configuration space. 0=MCA errors may be reported to the CPU 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 34 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 be set to 1 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 target or master abort error condition, and disables master abort and target abort reporting through the MCA error-reporting banks for PCI 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 is common during device enumeration).
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.
23:22	Reserved
21	**SyncOnAnyErrEn: sync flood on any error enable**. Read-write. 1=Enables flooding of the link with sync packets on detection of any MCA error that is uncorrectable, including link protocol errors.
20	**SyncOnWDTEn: sync flood on watchdog timer error enable**. Read-write. BIOS: 1. 1=Enables flooding of the link with sync packets on detection of a watchdog timer error.
19:18	Reserved.
17	**GenCrcErrByte1: generate CRC error on byte lane 1**. Read-Write. 1=Causes a CRC error to be injected on byte lane 1 of the link. The data carried by the link is unaffected. This bit is cleared after the error has been generated. This mechanism is independent of F0x84[ForceCrcErr]. In retry mode the per-packet CRC should be corrupted such that the corrupt bit(s) are sent on byte lane 1. See also F0x150[ForceErrType] for retry mode.
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. The data carried by the link is unaffected. This bit is cleared after the error has been generated. This mechanism is independent of F0x84[ForceCrcErr]. In retry mode the per-packet CRC should be corrupted such that the corrupt bit(s) are sent on byte lane 0. See also F0x150[ForceErrType] for retry mode.
15:14	Reserved.

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 = 1.28 ms 10b = 80 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 the outgoing link with sync packets when a sync packet is detected on the incoming link. Sync packets are propagated by default.
3	**SyncPktGenDis: sync packet generation disable**. Read-write. 1=Disables flooding of the outgoing link with sync packets when a CRC error is detected on the incoming link. By default, sync packet generation for CRC errors is controlled through [The Link Control Register] F0x84.
2	Reserved
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. 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
31:21	Reserved.
20:16	**ErrorCodeExt: extended error code**. Read-write. Logs the extended error code when an error is detected. See tables below for the error-code encoding.
15:0	**ErrorCode: error code**. Read-write. Logs an error code when an error is detected. See tables below for encoding.

Three types of errors are reported: TLB, memory, or bus errors.

Table 26: Error code types

Error Code	Error Code Type	Description

`0000 0000 0001 TTLL`	TLB	TT = Transaction Type
		LL = Cache Level

`0000 0001 RRRR TTLL`	Memory	Errors in the cache hierarchy (not in NB)
		RRRR = Memory Transaction Type
		TT = Transaction Type
		LL = Cache Level

`0000 1PPT RRRR IILL`	Bus	General bus errors including link and DRAM
		PP = Participation Processor
		T = Timeout
		RRRR = Memory Transaction Type
		II = Memory or IO
		LL = Cache Level

Table 27: Error codes: transaction type (TT)

TT	Transaction Type
00	Instruction
01	Data
10	Generic
11	Reserved

Table 28: Error codes: cache level (LL)

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

Table 29: Error codes: memory transaction type (RRRR)

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 30: Error codes: participation processor (PP)

PP	Participation Processor
00	Originated the request (SRC)
01	Responded to the request (RES)
10	Observed the error as a third party (OBS)
11	Generic

Table 31: Error codes: memory or IO (II)

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

The NB is capable of reporting the following errors.

Table 32: NB error descriptions

Error Type	Description	Control Bits (F3x40)
CRC Error	If the link is in retry mode, this may indicate excessive link reconnect failures; see F0x84[CrcErr, LinkFail, CrcFloodEn]. The NB will flood the outgoing link with sync packets after detecting a CRC error on an incoming link independent of the state of the control bits.	CrcErr0En
Sync Error	Link-defined sync error packets detected on link. The NB floods its outgoing link with sync packets after detecting a sync packet on an incoming link independent of the state of the control bits.	SyncPkt0En

Table 32: NB error descriptions

Error Type	Description	Control Bits (F3x40)
Mst Abort	Master abort seen as result of link operation. Reasons for this error include requests to non-existent addresses. The NB returns an error response back to the requestor with any associated data all 1s independent of the state of the control bit.	MstrAbortEn
Tgt 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
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. Error signaled back to requester. Cause of error may be another processor or device which failed to respond.	WDTRptEn
DEV Error	SVM DEV error detected.	DevErrEn
Link Data Error	Data error detected on link.	HtDataEn, McaUsPwDatErrEn
Link Protocol Error	Protocol error detected on link. Ensure that error is not due to failure or reset at far end of link or from transmission corruption (should be 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.	HtProtEn
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
DEV Table Walk Data Error	An uncorrectable error was found in data returned from a DEV table walk.	TblWlkDatErrEn

Table 33: NB error signatures, part 1

Error Type	Error Threshold Group	Ext. Error	Error Code (see F3x48 for encoding)					
			Type	PP	TT	RRRR	II/TT	LL
Reserved.	-	0_0000	-	-	-	-	-	-

Table 33: NB error signatures, part 1

Error Type	Error Threshold Group	Ext. Error	Error Code (see F3x48 for encoding)					
			Type	PP	TT	RRRR	II/TT	LL
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	LG
Tgt Abort		0_0100	BUS	SRC/OBS	0	RD/WR	MEM/IO	LG
RMW Error		0_0110	BUS	OBS	0	GEN	MEM/IO	LG
WDT Error		0_0111	BUS	GEN	1	GEN	GEN	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
Link Protocol Error		0_1011	BUS	OBS	0	GEN	GEN	LG
Link Retry		0_1110	BUS	OBS	0	GEN	GEN	LG
DEV Table Walk Error		0_1111	BUS	OBS	0	GEN	MEM	LG

Table 34: NB error signatures, part 2

Error Type	[The MCA NB Status High Register] F3x4C settings							
	UC	AddrV	PCC	Syndrome Valid	Reserved	Reserved	LDT Link	Err CPU
CRC Error	1	0	1	-	0	0	Y	-
Sync Error	1	0	1	-	0	0	Y	-
Mst Abort	1	1	If CPU source	-	0	0	Y	Y
Tgt Abort	1	1	If CPU source	-	0	0	Y	Y
RMW Error	1	1	0	-	0	0	Y	-
WDT Error	1	1[1]	1	-	0	0	-	-
DEV Error	1	1	0	-	0	0	Y	-
Link Data Error	1	1	0	-	0	0	Y	-
Link Protocol Error	1	0[2]	1	-	0	0	Y	-
Link Retry	0	0	0	-	0	0	Y	-
DEV Table Walk Error	1	1	0	-	0	0	-	-

1. See [The MCA NB Address Low Register encoding for Watchdog Timer Errors] Table 36:
2. See [The MCA NB Address Low Register encoding for Link Protocol Errors] Table 35:

F3x4C MCA NB Status High Register

Cold Reset: xxxx xxxxh.
Software is normally only allowed to write 0's to this register to clear the fields so subsequent errors may be logged. See 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	**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].
29	**UC: error uncorrected**. Read-write; set-by-hardware. 1=The error was not corrected by hardware.
28	**En: error enable**. Read-write; set-by-hardware. 1=The MCA error reporting is enabled for this error in the MCA Control Register.
27	Reserved.
26	**AddrV: error address valid**. Read-write; set-by-hardware. 1=The address saved in the address register is the address where the error occurred.
25	**PCC: processor context corrupt**. Read-write; set-by-hardware. 1=The state of the processor may be corrupted by the error condition. Reliable restarting might not be possible.
24:10	Reserved
9	**SubLink: sublink**. Read-write; set-by-hardware. Indicates if the error was associated with the upper or lower byte of the link. 0=Bits[7:0], 1=Bits[15:8].
8	Reserved.
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[3:1]:Reserved. LDTLink[0]:Error associated with link 0.
3:0	**ErrCPU[3:0]: error associated with CPU N**. Read-write; set-by-hardware. This field indicates which core within the processor is associated with the error. ErrCPU[3:2]: Reserved. ErrCPU[1] = Error associated with core 1. ErrCPU[0] = Error associated with core 0.

F3x50 MCA NB Address Low Register

Cold Reset: xxxx xxxxh. Read-write. [The MCA NB Address Low Register] F3x50 and [The MCA NB Address High Register] F3x54 carry the address associated with a machine check error, other fields, or both. See the following tables for the format for each error type.

Table 35: MCA NB Address Low Register encoding for Link Protocol Errors

ErrAddr Bits	Value	Description
63:5	-	Reserved.

Table 35: MCA NB Address Low Register encoding for Link Protocol Errors

ErrAddr Bits	Value	Description
4:1	0000b	Read Response without matching request. Used for any of the following cases: • Read Response received without matching request with same SrcTag. • Read Response received in response to WrSized request. • Read Response received, but the Count field doesn't match original RdSized request.
	0001b	Reserved.
	0010b	TgtDone without matching request. This encoding is used for any of these cases: • TgtDone received without matching request with same SrcTag. • TgtDone received in response to RdSized.
	0011b	Reserved.
	0100b	Command buffer overflow.
	0101b	Data buffer overflow.
	0110b	Link retry packet count acknowledge overflow.
	0111b	Data command in the middle of a data transfer.
	1000b	Link address extension command followed by a packet other than a command with address.
	1001b	Reserved.
	1010b	A command with invalid encoding was received. This error occurs when: (1) any invalid command is received while not in retry mode or (2) any illegal command is received in which the CRC is correct while in retry mode.
	1011b	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).
	1100b	Link CTL timeout.
	1101b	Reserved.
	111xb	Reserved.
0	-	Reserved.

Table 36: MCA NB Address Low Register encoding for Watchdog Timer Errors

ErrAddr Bits	Description
63:60	RAQ Wait code (0000b means the op was not waiting in RAQ): • [63]=1 means all inbound data has not been transferred across ONION. • [62]=1 means ordering rules not satisfied for Dispatch of Response. • [61]: Reserved. • [60]=1 means lack of ONION downstream credits.
59	Priority VC set.

Table 36: MCA NB Address Low Register encoding for Watchdog Timer Errors

<table>
<tr><th>ErrAddr Bits</th><th>Description</th></tr>
<tr><td>58:54</td><td>Packet Routing
<table>
<tr><th>Value</th><th>Requester</th><th>Target</th></tr>
<tr><td>00000b</td><td>Cpu0</td><td>DRAM</td></tr>
<tr><td>00100b</td><td>Cpu1</td><td>DRAM</td></tr>
<tr><td>01000b</td><td>Link</td><td>DRAM</td></tr>
<tr><td>10000b</td><td>DEV TblWlk</td><td>DRAM</td></tr>
<tr><td>00001b</td><td>Cpu0</td><td>INT*</td></tr>
<tr><td>00101b</td><td>Cpu1</td><td>INT*</td></tr>
<tr><td>01001b</td><td>Link</td><td>INT*</td></tr>
<tr><td>00010b</td><td>Cpu0</td><td>Link</td></tr>
<tr><td>00110b</td><td>Cpu1</td><td>Link</td></tr>
<tr><td>01010b</td><td>Link</td><td>Link</td></tr>
</table>
All other values are reserved.
*INT = Internal target; one of register, APIC, SysMgt.</td></tr>
<tr><td>53:48</td><td>Link command.</td></tr>
<tr><td>47:40</td><td>Reserved.</td></tr>
<tr><td>39:2</td><td>Error address.</td></tr>
<tr><td>1</td><td>Reserved.</td></tr>
<tr><td>0</td><td>Request was dispatched.</td></tr>
</table>

F3x54 MCA NB Address High Register

Cold Reset: xxxx xxxxh. 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 2.6.4 [Memory Scrubber]. The scrub rate is specified as the time between successive scrub events. A scrub event occurs when a line of cache memory is checked for errors; the amount of cache memory that is checked varies based on the cache 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	1.31072 ms
01h	Reserved	09h	10.24 us	11h	2.62144 ms
02h	Reserved	0Ah	20.48 us	12h	5.24288 ms
03h	Reserved	0Bh	40.96 us	13h	10.4858 ms
04h	Reserved	0Ch	81.92 us	14h	20.9715 ms
05h	Reserved	0Dh	163.84 us	15h	41.9430 ms
06h	1.28 us	0Eh	327.68 us	16h	83.8861 ms
07h	2.56 us	0Fh	655.36 us	All others - reserved.	

Bits	Description
31:21	Reserved.
20:16	**DcacheScrub: data cache scrub rate**. Read-write. Specifies time between 64-bit data cache scrub events.

15:13	Reserved.
12:8	**L2Scrub: L2 cache scrub rate**. Read-write. Read-write. Specifies time between 64-byte L2 scrub events.
7:0	Reserved.

F3x64 Hardware Thermal Control (HTC) Register

See section 2.10.4.1 [Hardware Thermal Control (HTC) and PROCHOT_L] for information on HTC. F3x64 is reserved if F3xE8[HtcCapable]==0. If (F3xE8[HtcCapable]) THEN

Bits	Description
31	**HtcLock: HTC lock**. Read; write-1-only; Reset: 0. 1=HtcPstateLimit, HtcHystLmt, HtcTmpLmt, and HtcEn are read-only. 0=HtcPstateLimit, HtcHystLmt, HtcTmpLmt, and HtcEn are read-write.
30:28	**HtcPstateLimit: HTC P-state limit select**. Read-write. Reset state varies by product. Specifies the P-state limit of all cores when in the HTC-active state. The HtcPstateLimit to apply is not changed if the value of this field is greater than MSRC001_0061[PstateMaxVal]. See also section 2.10.4.1 [Hardware Thermal Control (HTC) and PROCHOT_L].
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.

ELSE // F3xE8[HtcCapable]==0

Bits	Description

31:0	Reserved.

ENDIF

F3x6C Upstream ONION Data Buffer Count Register

See section 2.6.3.2.1 [Display Refresh And IFCM].

Bits	Description
31:28	Reserved.
27:24	**UpHiRespDBC: upstream high priority response data buffer count**. Read-only. Reset: 8h.
23:20	**UpHiNpreqDBC: upstream high priority non-posted request data buffer count**. Read-write; changes take effect on next warm reset. Cold reset: 0h.
19:16	**UpHiPreqDBC: upstream high priority posted request data buffer count**. Read-write; changes take effect on next warm reset. Cold reset: 0h.
15:12	Reserved.
11:8	**UpLoRespDBC: upstream low priority response data buffer count**. Read-only. Reset: 8h.
7:4	**UpLoNpreqDBC: upstream low priority non-posted request data buffer count**. Read-write; changes take effect on next warm reset. Cold reset: 2h.
3:0	**UpLoPreqDBC: upstream low priority posted request data buffer count**. Read-write; changes take effect on next warm reset. Cold reset: 6h.

F3x74 Upstream ONION Command Buffer Count Register

Read-write; changes take effect on next warm reset. See section 2.6.3.2.1 [Display Refresh And IFCM].

Bits	Description
31:28	**UpDRReqCBC: upstream display refresh non-posted command buffer count**. Cold reset: 0.
27:24	**UpHiRespCBC: upstream high priority response command buffer count**. Cold reset: 0.
23:20	**UpHiNpreqCBC: upstream high priority non-posted command buffer count**. Cold reset: 0.
19:16	**UpHiPreqCBC: upstream high priority posted command buffer count**. Cold reset: 0.
15:12	Reserved.
11:8	**UpLoRespCBC: upstream low priority response command buffer count**. Cold reset: 8h.
7:4	**UpLoNpreqCBC: upstream low priority non-posted command buffer count**. Cold reset: Ah.
3:0	**UpLoPreqCBC: upstream low priority posted command buffer count**. Cold reset: 6h.

F3x7C In-Flight Queue Buffer Allocation Register

Read-write; changes take effect on next warm reset. See section 2.6.3.2.1 [Display Refresh And IFCM].

Bits	Description
31:28	**FreePoolBC: free pool buffer count**. Cold reset: Dh. This field must be at least 1.
27:24	**DispRefBC: display refresh buffer count**. Cold reset: 0h.
23:20	**HiPriNPBC: high priority channel non-posted buffer count**. Cold reset: 0h.

19:16	**HiPriPBC: high priority channel posted buffer count**. Cold reset: 0h.
15:12	**LoPriNPBC: low priority channel non-posted buffer count**. Cold reset: 1h.
11:8	**LoPriPBC: low priority channel posted buffer count**. Cold reset: 1h.
7:4	Reserved.
3:0	**CpuBC: CPU buffer count**. Cold reset: 1h.

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 2.4.2 [P-states] for information on which states are supported. See “ACPI Power State Transitions” on page 35.

Table 37: SMAF Code Definition

SMAF [7:0]	Transaction Type
7:5	**CpuDid: clock divisor ID**. Read-write. • If MSRC001_00[6B:64][CpuDid] of the current P-state is greater than or equal to F3x[84:80][CpuDid], then no frequency change is made when entering the low-power state associated with this register. • The resultant frequency is: (the frequency specified by F3xD4[MainPllOpFreqId, MainPllOpFreqIdEn]) / (the divisor specified by CpuDid). • CpuDid 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 2.6.4 [Memory Scrubber].
4	**CpuAltVidEn: alternate VID enable**. Read-write. 1=The altvid (F3xDC[AltVid]) is driven to the core power planes while in the low-power state. 0=The altvid (F3xDC[AltVid]) is not driven to the core power planes while in the low-power state.
3	**DramDllVRegPwrDwn: DRAM DLL voltage regulator power down enable**. Read-write. See F2x[1,0]A0[DllPwrDwn, DllVRegPwrDwn]. DramSr is required to be set if this bit is set.
2	**DramMemClkTri: DRAM memory clock tristated**. Read-write. 1=DRAM MEMCLK is tristated while in the low-power state. DramSr is required to be set if this bit is set.
1	**DramSr: DRAM self-refresh**. Read-write. 1=DRAM is placed into self-refresh while in the low-power state.

Table 37: SMAF Code Definition

<table>
<tr><th>SMAF [7:0]</th><th>Transaction Type</th></tr>
<tr><td>0</td><td><b>CpuPrbEn: CPU direct probe enable</b>. Read-write. Specifies how probes are handled while in the low-power state and varies as a function of CpuAltVidEn. This bit should be set if probes are expected to occur while in the low-power state associated with the SMAF.
<table>
<tr><th>CpuAltVidEn</th><th>CpuPrbEn</th><th>Definition</th></tr>
<tr><td>x</td><td>0</td><td>The core COF and VID is transitioned to the current P-state and all outstanding probes are completed. When the last probe has been completed and the hysteresis time (F3xD4[ClkRampHystSel]) has elapsed, the COF is transitioned to F3x[84:80][CpuDid].</td></tr>
<tr><td>0</td><td>1</td><td>The core COF does not change and all outstanding probes are handled at the frequency specified by CpuDid; this mode is only supported if CpuDid is <= 100b (<= 16).</td></tr>
<tr><td>1</td><td>1</td><td>The core COF and VID are transitioned to Pmin (F3xDC[PstateMaxVal]) and all outstanding probes are completed. When the last probe has been completed the COF is transitioned to F3x[84:80][CpuDid] when LDTSTOP asserts, without waiting the hysteresis time.</td></tr>
</table>
</td></tr>
</table>

Table 38: F3x84 ACPI Power State Control Register High

Bits	SMAF Code	ACPI State	Setting	Description
31:24	111b	C1	80h	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. This field has no function if F3xD4[Smaf7Dis]=1.
23:16	110b	S4/S5	E6h	Initiated by a processor access to the ACPI-defined PM1_CNTa register.
15:8	101b	N/A	00h	Reserved.
7:0	100b	S3	EEh	Initiated by a processor access to the ACPI-defined PM1_CNTa register.

Table 39: F3x80 ACPI Power State Control Register Low

Bits	SMAF Code	ACPI State	Setting	Description
31:24	011b	S1	FFh	Initiated by a processor access to the ACPI-defined PM1_CNTa register.
23:16	010b	N/A	00h	Reserved.
15:8	001b	C1E, or Link init	(See Description on right)	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. The C1E state is not supported when CLMC is used by the IO Hub. Setting: • Discrete graphics: BIOS: {F3x1F0[ClkDivC1E[2:0]],11111b}. • UMA graphics: BIOS: {F3x1F0[ClkDivC1E[2:0]],10011b}.
7:0	000b	N/A	00h	Reserved.

F3xA0 Power Control Miscellaneous Register

Bits	Description
31	Reserved.
30	**VddCpuGanged: VDD for all cores are ganged on motherboard**. Read-write. Reset: 0. 1=VDD for all cores are ganged in the motherboard (dual-plane operation). VDDNB is separate and independently controllable from the ganged CPU VDD plane on the motherboard. The SVI voltage sent for Vdd0 and Vdd1 is the maximum voltage specified for the core(s). 0=VDD for all cores, and VDDNB are all separate independently controllable voltage planes on the motherboard. (triple-plane operation) • BIOS must initialize VddCpuGanged prior to a software initiated P-state change (writing MSRC001_0062[PstateCmd]) or enabling HTC (F3x64[HtcEn]=1).
29:28	Reserved.
27:16	**PstateId: P-state identifier**. Read-only. This field specifies the P-state ID associated with the product.
15:10	Reserved.
9	**SviHighFreqSel: SVI high frequency select**. Read-write. Cold reset: 0. BIOS: 1. 0=400 KHz. 1=3.4 MHz. Writes to this field take effect at the next SVI command boundary.
8	Reserved.
7	**PsiVidEn: PSI_L bit VID enable**. Read-write. Reset: 0. This bit specifies how the PSI_L bit 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 bit is as specified by the PsiVid field of this register. 0=The PSI_L bit is always high. See section 2.4.1.4 [PSI_L Bit].
6:0	**PsiVid: PSI_L bit VID threshold**. Read-write. Reset: 0. When enabled by PsiVidEn, this field specifies the threshold value of VID code generated by the processor, which in turn determines the state of the PSI_L bit. 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 the PSI_L bit is high; when the VID code is greater than or equal to PsiVid, the PSI_L bit is driven low. See 2.4.1.4 [PSI_L Bit].

F3xA4 Reported Temperature Control Register

The processor measures temperature to 1-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[10:0]: 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].
20:13	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	Reserved.
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 ms, ranging from 1 to 8 ms. 01b (Bits[2:0] + 1) * 10 ms, ranging from 10 to 80 ms. 10b (Bits[2:0] + 1) * 100 ms, ranging from 100 to 800 ms. 11b (Bits[2:0] + 1) * 1 second, ranging from 1 to 8 seconds.

F3xD4 Clock Power/Timing Control 0 Register

Bits	Description
31:19	Reserved.
18	**Smaf7Dis: SMAF 7 (C1) disable**. Read-write. Reset: 0. 1=Disable F3x84[31:24] and enable F4x174_x[0F:00][LmmCpuDid[2:0], LmmCpuAltVidEn, LmmDramMemClkTri, LmmDramSr, LmmCpuPrbEn]. 0=Enable F3x84[31:24] and disable F4x174_x[0F:00][LmmCpuDid[2:0], LmmCpuAltVidEn, LmmDramMemClkTri, LmmDramSr, LmmCpuPrbEn]. See F3x[84:80].
17:16	**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. Bits PLL lock time 00b 1 us. 01b 10 us. 10b 100 us. 11b 1000 us.
15:12	Reserved.
11:8	**ClkRampHystSel: clock ramp hysteresis select**. Read-write. Cold reset: 000b. BIOS: 2h. 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. 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. Values range from 0h=320ns to Fh=5120ns, encoded as 320ns * (1 + ClkRampHystSel).

7	Reserved.
6	**MainPllOpFreqIdEn: main PLL operating frequency ID enable**. Read-write; changes take effect on next warm reset. Cold reset: 1. This specifies the main PLL operating frequency after a warm or cold reset. 1=The main PLL operating frequency is specified by F3xD4[MainPllOpFreqId]. 0=The main PLL operating frequency is 1.6 GHz, regardless of the state of F3xD4[MainPllOpFreqId].
5:0	**MainPllOpFreqId: main PLL operating frequency ID**. Read-write; changes take effect on next warm reset. Cold reset: value varies by product. • See F3xD4[MainPllOpFreqIdEn]. • F3xD4[MainPllOpFreqId] is defined as follows: 100 MHz * (F3xD4[MainPllOpFreqId] + 08h). • If (MSRC001_0071[MainPllOpFreqIdMax]=0) then there is no logical frequency limit. • If (MSRC001_0071[MainPllOpFreqIdMax]>0) then programming F3xD4[MainPllOpFreqId] >MSRC001_0071[MainPllOpFreqIdMax] results in a frequency equal to MSRC001_0071[MainPllOpFreqIdMax].

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. The VDD VID(s) can also be automatically controlled when entering low-power states, as specified by [The ACPI Power State Control Registers] F3x[84:80][CpuAltVidEn].

If the alternate VID is applied, then the exit sequence includes: (1) the processor applies the prior operational VID, (2) the processor waits for the time specified by AltVidSlamTime, and (3) the processor allows instruction execution to continue. When entering such low-power states, core clocks are ramped down prior to applying the alternate VID.

Bits	Description
31:28	Reserved.
27:24	**ReConDel: link reconnect delay**. Read-write. Cold reset: 0. BIOS: 3h. 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 F0x84[LdtStopTriEn]=1 and F0x170[RxLSSel]!=(LS0 or LS1). The assertion of CTL is delayed until the specified time has elapsed. See section 2.7.5 [Link LDTSTOP_L Disconnect-Reconnect] for information on when this is applied. The receiver always enabled 1us after deassertion of LDTSTOP_L, regardless of the setting of this field or other delays in assertion of CTL. 0h 1us. 1h 2us. ... ... 9h 10us. Fh-Ah Reserved.
23:7	Reserved.
6:4	**AltVidSlamTime: voltage stabilization slam time**. Read-write. Cold reset: 111b. BIOS: 010b. Specifies the time to wait for voltage stabilization between an SVI command that increases the voltage from altvid to Pmin and the FID change. See also section 2.4.1.8 [Hardware-Initiated Voltage Transitions]. Bits Time Bits Time 000b 10 us 100b 60 us 001b 20 us 101b 100 us 010b 30 us 110b 200 us 011b 40 us 111b 500 us

3	Reserved.
2:0	**VSSlamTime: voltage stabilization slam time**. Read-write. Cold reset: 111b. Specifies the time to wait for voltage stabilization between an SVI command that changes the voltage and the FID change, except for altvid to Pmin voltage transitions. See AltVidSlamTime. There is no wait time between an SVI command that decreases the voltage and the FID change. See also section 2.4.1.8 [Hardware-Initiated Voltage Transitions]. • BIOS: VSSlamTime time >= (P0 voltage – altvid voltage) / (5mV/us), where the P0 voltage is specified by MSRC001_0064[CpuVid] and the altvid voltage is specified by F3xDC[AltVid]. Bits Time Bits Time 000b 10 us 100b 60 us 001b 20 us 101b 100 us 010b 30 us 110b 200 us 011b 40 us 111b 500 us

F3xDC Clock Power/Timing Control 2 Register

Bits	Description
31:19	Reserved.
18:12	• **NbVid**. This specifies the NB VID. Read-write. Cold reset: value varies by product. See section 2.4.1 [Processor Power Planes And Voltage Control]. VDDNB will be changed if the written value for F3xDC[NbVid] != F3xDC[NbVid].
11	Reserved.
10:8	**PstateMaxVal: P-state maximum value**. Read-write. Cold reset: specified by the reset state of MSRC001_00[6B:64][PstateEn]; the cold reset value is the highest P-state number corresponding to the MSR in which PstateEn is set (e.g., if MSRC001_0064 and MSRC001_0065 have this bit set and the others do not, then 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	Reserved.
6:0	**AltVid: alternate VID**. Read-write. Cold reset: value varies by product. Specifies the VID driven to the VDD power plane(s) while in the low-power state, as specified by F3x[84:80][CpuAltVidEn]. This field is required to be programmed as specified by MSRC001_0071[MaxVid and MinVid]. See section 2.4.1 [Processor Power Planes And Voltage Control].

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 to the thermal diode connected to pins. See also section 2.10.2 [Thermal Diode]. It is encoded as follows: 00h is undefined. 01h to 3Fh: correction = +11C - DiodeOffset = {+10C to -52C}. 40h to 7Fh: undefined.
7:6	Reserved.
5	**ThermtpEn: THERMTRIP enable**. Read-only. 1=The THERMTRIP state as specified in section 2.10.4.2 [THERMTRIP] is supported by the processor.
4	Reserved.
3	**ThermtpSense: THERMTRIP** sense. Read-only. Cold reset: 0. 1=The processor temperature exceeded the THERMTRIP value (regardless as to whether the THERMTRIP state (ThermtpEn) is enabled). This bit is also set when the diagnostic bit SwThermtp = 1.
2	Reserved.
1	**Thermtp: THERMTRIP**. Read-only. 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
31:14	Reserved.
13:12	**CmpCap: CMP capable**. Specifies the number of cores enabled on the device. 00b=1; 01b=2; 10b=Reserved; 11b=Reserved.
11	**Link error-retry capable**.
10	**HtcCapable: HTC capable**.
9	**SvmCapable: SVM capable**.
8	**MctCap: memory controller (on the processor) capable**. Reset: 1.
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 Reserved 101b 667 MT/s 010b Reserved 110b 533 MT/s 011b 1067 MT/s 111b 400 MT/s
4:1	Reserved.
0	**DctDualCap: two-channel DRAM capable (i.e., 128 bit)**. 0=Single channel (64-bit) only.

F3xF0 DEV Capability Header Register

See section 2.6.2 [DMA Exclusion Vectors (DEV)]. Note: if SVM is not supported, as specified by F3xE8[SVM Capable], then this register is reserved.

DMA Exclusion Vectors (DEV) are contiguous arrays of bits in physical memory. There is no support for

MMIO DEV tables. Each bit in the DEV table represents a 4KB page of physical memory; the DEV applies to accesses that target system memory and MMIO. The DEV table is packed as follows: bit[0] of byte 0 (pointed to by the DEV table base address, F3xF8_DF0 and F3xF8_DF1) controls the first 4K bytes of physical memory (starting at address 00_0000_0000h); bit[1] of byte 0 controls the second 4K bytes of physical memory; etc. When a DEV table bit is set to one, accesses to that physical page by external DMA devices is not allowed. If an external device attempts to access a protected physical page, then the processor master aborts the request.

In addition, the processor supports multiple protection domains. There is a DEV table for each protection domain. Link-defined UnitIDs or RequesterIDs may be assigned to the DEV of a specific protection domain through F3xF8_DF2. 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_DF[7:0], which are defined following F3xF8. [The DEV Function/Index Register] F3xF4 and [The DEV Data Port] F3xF8 are used to access F3xF8_DF[7:0]. The register number (i.e., the number that follows "_DF" in the register mnemonic) is specified by F3xF4[DevFunction]. In addition, F3xF8_DF0, F3xF8_DF1, and F3xF8_DF2 are each instantiated multiple times, indexed by F3xF4[DevIndex]. Access to these registers is accomplished as follows:

- Reads:
  - Write the register number to F3xF4[DevFunction,DevIndex].
  - Read the register contents from F3xF8.

- Writes:
  - Write the register number to F3xF4[DevFunction,DevIndex].
  - Write the register contents to F3xF8.

Bits	Description
31:22	Reserved.
21	**IntCap: interrupt reporting capability**. Read-only. Reset: 0. 0=Interrupt reporting of DEV protection violations is not present on this device.
20	**MceCap: MCE reporting capability**. Read-only. Reset: 1. Indicates that machine check architecture reporting of DEV protection violations is present on this device.
19	Reserved.
18:16	**CapType: DEV capability block type**. Read-only. Reset: 000b. Specifies the layout of the Capability Block.
15:8	**CapPtr: capability pointer**. Read-only. Reset: 00h. Indicates that this is the last capability block.
7:0	**CapId: capability ID**. Read-only. Reset: 0Fh. Indicates a DEV capability block.

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. Writing invalid values may result in undefined behavior.

7:0	**DevIndex**. Read-write. See F3xF0 for details. Valid values for this field are (1) 00h through (F3xF8_DF3[NDomains] - 1) when either F3xF8_DF0 or F3xF8_DF1 are being accessed and (2) 00h through (F3xF8_DF3[NMaps] + F3xF8_DF3[NSrcMaps] - 1) when F3xF8_DF2 is being accessed; this field is reserved for accesses to all other DEV configuration registers. Writing invalid values may result in undefined behavior.

F3xF8 DEV Data Port

Note: if SVM is not supported, as specified by F3xE8[SVM Capable], then this location and registers F3xF8_DF[7:0] are reserved. See F3xF0 for details about this port.

F3xF8_DF0 DEV Base Address/Limit Low Register

Reset: 0000 0000h.

This register is instantiated multiple times, specified by F3xF8_DF3[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_DF1[BaseAddress[47:32]] to specify the base address of the DEV table. The DEV table is required to be in either non-cacheable or write-through memory. Placing DEV tables in MMIO space is not supported. If any part of the DEV table is in other than system memory, then undefined behavior results.
11:7	Reserved.
6:2	**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). 00h 4 GB 01h 8 GB ... ... 08h 1024 GB 09h-1Fh Reserved.
1	**Protect: protect out-of-range addresses**. Read-write. 0=DMA accesses to addresses that are outside the range covered by the DEV table are allowed. 1=DMA accesses to addresses that are outside the range covered by the DEV table are protected.
0	**Valid: DEV table valid**. Read-write. 1=The DEV table for the protection domain specified by 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_DF1 DEV Base Address/Limit High Register

Reset: 0000 0000h.

This register is instantiated multiple times, specified by F3xF8_DF3[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
31:8	Reserved.

7:0	**BaseAddress[39:32]: DEV table base address bits[39:32]**. Read-write. See F3xF8_DF0[BaseAddress].

F3xF8_DF2 DEV Map Register

Reset: 0000 0000h.

The DEV Map register has two formats. The first format, called the UID format, is the existing format defined for mapping link Unit IDs to DEV protection domains. The second format, called the BDF format, shown below is used for mapping requester IDs (BDF) to DEV protection domains.

This register is instantiated multiple times, equal to F3xF8_DF3[NMaps] + F3xF8_DF3[NSrcMaps]. See F3xF0 for more details. F3xF4[DevIndex], from 0 to F3xF8_DF3[NMaps]-1, is the UID format. F3xF4[DevIndex], from F3xF8_DF3[NMaps] to (F3xF8_DF3[NMaps]+F3xF8_DF3[NSrcMaps]-1), is the BDF format.

For the UID format, if Valid[x] is set, then the address of DMA request received by the processor from an IO link of bus number BusNu and with a UnitID of Unit[x] are checked against the DEV table of protection domain number Dom[x] to determine if the transaction is allowed. A UnitID can only be assigned to one protection domain. If a UnitID is assigned to more than one protection domain the results are undefined.

For the BDF format, if Valid is set, then the address of DMA request received by the processor from an IO link with a BDF of RequesterID are checked against the DEV table of protection domain number Domain to determine if the transaction is allowed. If the BDF 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 BDF can only be assigned to one protection domain. If a BDF is assigned to more than one protection domain the results are undefined.

In the request matches on both a UID format map register and a BDF format map register and both map registers specify different domains, the domain of the BDF format register dominates.

If the request doesn’t match on any map register then the address of the request is checked against the DEV table of protection domain 0 to determine if the transaction is allowed.

Table 40: F3xF8_DF2: DEV Map UID Format

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.

Table 41: F3xF8_DF2: DEV Map BDF Format

Bits	Description

Table 41: F3xF8_DF2: DEV Map BDF Format

31:16	**RequesterID: requester bus, device, and function**. Read-write.
15:10	**Domain: protection domain**. 3 LSBs are read-write; 3 MSBs are read-only, 000b. This is the protection domain number assigned to RequesterID.
9:1	Reserved.
0	**Valid: requester ID valid**. Read-write. 1=Enable DEV checking for RequesterID and Domain.

F3xF8_DF3 DEV Capabilities Register

Bits	Description
31:24	**NSrcMaps: number of map registers implemented**. Read-only, 08h. Specifies the number of instantiations of F3xF8_DF2 of the BDF format.
23:16	**NMaps: number of map registers implemented**. Read-only, 04h. Specifies the number of instantiations of F3xF8_DF2 of the UID format.
15:8	**NDomains: number of protection domains implemented**. Read-only, 08h. Specifies the number of protection domains and the number of instantiations of F3xF8_DF0 and F3xF8_DF1.
7:0	**Revision: DEV register-set revision number**. Read-only, 01h. 01h=Support provided for F3xF8_DF2 BDF format.

F3xF8_DF4 DEV Control Register

Reset: 0000 0000h.

Bits	Description
31:8	Reserved.
7	**SrcMapEn: source map enable**. Read-write. 1=If a request includes a source identification packet, only the Requester ID map registers are used to determine the domain. If a request does not includes a source identification packet, only the UnitID map registers are used to determine the domain. 0=The UnitID map registers are used to determine the domain for all requests.
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; cleared-by-software. This bit is set by hardware after an SKINIT instruction. 1=The memory region associated with the SKINIT instruction is protected from DMA access. Software writing this bit to a 1 does not have any affect.
4	**DevInv: invalidate DEV cache**. Read; write-1-only; cleared-by-hardware. 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. 1=Upstream IO-space accesses are regarded as DEV protection violations.
1	Reserved.
0	**DevEn: DEV enable**. Read-write. 1=Enables DMA exclusion vector protection.

F3xF8_DF5 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_DF6 and F3xF8_DF7.

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_DF6 and F3xF8_DF7 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) 100b = IO space 001b = DRAM 101b = Configuration 010b = MMIO 110b = reserved 011b = 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) 010b = IO device 001b = CPU 011b - 111b = reserved
1:0	**ErrCodeAccType: error code access type**. Read-write; set-by-hardware. Specifies the access type of the transaction that resulted in the protection violation. 00b = Generic (or could not be determined) 10b = Write 01b = Read 11b = Read-modify-write

F3xF8_DF6 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_DF5 for details.
1:0	Reserved.

F3xF8_DF7 DEV Error Address High Register

Cold reset: 0000 0000h.

Bits	Description
31:8	Reserved.
7:0	**ErrAddr: error address bits[39:32]**. Read-write; set-by-hardware. See F3xF8_DF5 for details.

F3xFC CPUID Family/Model Register

These values are identical to the values read out through CPUID Fn0000_0001_EAX and CPUID Fn8000_0001_EAX; see that register for details.

Bits	Description
31:28	Reserved. Reset: 0h.
27:20	**ExtFamily: extended family**. Read-only. Reset: 2h.
19:16	**ExtModel: extended model**. Read-only. Reset value varies by product.
15:12	Reserved. Reset: 0h.
11:8	**BaseFamily**. Reset: Fh.
7:4	**BaseModel**. Read-only. Read-only. Reset value varies by product.
3:0	**Stepping**. Read-only. Reset value varies by product.

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 table walker encounters 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.
21:9	Reserved.
8	**SyncOnHtProtEn: sync flood on link protocol error enable**. Read-write. 1=Enables sync flood on detection of a protocol error on a link.
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.
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. This bit should be set to prevent bad data from propagating from the processor.
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).

F3x188 NB Extended Configuration Register

Bits	Description
31:28	**FeArbCpuWeightOverHiPrio[3:0]: NB front-end round-robin arbiter CPU weight over high priority channel**. Read-write. Reset: 4h. BIOS: 1h. This value is the number of times (out of 16) that the arbiter favors the CPU when both a CPU and the High Priority channel are requesting. F3x188[FeArbCpuWeightOverLoPrio] must be >= F3x188[FeArbCpuWeightOverHiPrio]. Note: When two CPUs are enabled each CPU is granted half of this value on average.
27:24	**FeArbCpuWeightOverLoPrio[3:0]: NB front-end round-robin arbiter CPU weight over low priority channel**. Read-write. Reset: Bh. BIOS: Bh. This value is the number of times (out of 16) that the arbiter favors the CPU when both a CPU and the Low Priority channel are requesting. F3x188[FeArbCpuWeightOverLoPrio] must be >= F3x188[FeArbCpuWeightOverHiPrio]. Note: When two CPUs are enabled each CPU is granted half of this value on average.
23	**EnCpuSerRdBehindIoRd: enable CPU serialization of reads behind IO reads**. Read-write. Reset: 0. BIOS: 0. 1=From the same CPU, sized reads are serialized behind IO reads. 0=From the same CPU, sized reads are not serialized behind IO reads.
22	**EnCpuSerRdBehindNpIoWr: enable CPU serialization of reads behind non-posted IO writes**. Read-write. Reset: 0. BIOS: 0. 1=From the same CPU, sized reads are serialized behind non-posted IO writes. 0=From the same CPU, sized reads are not serialized behind non-posted IO writes.
21	**EnCpuSerWrBehindIoRd: enable CPU serialization of writes behind IO reads**. Read-write. Reset: 0. BIOS: 0. 1=From the same CPU, sized writes are serialized behind IO reads. 0=From the same CPU, sized writes are not serialized behind IO reads.
20:13	Reserved.
12:10	**DlyDropOnionSyncInC3: delay dropping onion synchronizer when entering C3**. Read-write. Reset: 0. BIOS: 001b. The delay from when all conditions are met to enter C3 to when the NCLK frequency is divided. 000b = 0 NCLKs 001b = 100 NCLKs 010b = 200 NCLKs 011b = 400 NCLKs 100b = 800 NCLKs 101b = 1600 NCLKs 110b = 3200 NCLKs 111b = 6400 NCLKs
9:8	Reserved.
7	**DisNonIsocThrottleForDR: throttling of non-isoc ops is disabled**. Read-write. Reset: 0. BIOS: 0. 1=Throttling of non-Isoc ops is disabled. 0=Throttling of non-Isoc ops is enabled (if write throttling is active).
6:5	**DramWrThrottleStopModeForDR: disable DRAM write throttling for display refresh**. Read-write. Reset: 00b. Specifies when to stop write throttling after it has started. Bits Definition 00b After 256 NCLKs. 01b After 64 NCLKs. 10b After 16 NCLKs. 11b When there are no DR ops pending in inbound ONION queue. • BIOS: 01b if !(F2x[1,0]90[BurstLength32] && (F2x[1,0]94[MemClkFreq]==000b)). • BIOS: 00b if (F2x[1,0]90[BurstLength32] && (F2x[1,0]94[MemClkFreq]==000b)).

4:3	**DramWrThrottleStartModeForDR : disable DRAM write throttling for display refresh**. Read-write. Reset: 10b. BIOS: 11b. Specifies when to start write throttling. Bits Definition 00b Write throttling is disabled. 01b At least 1 DR op is pending in the inbound ONION queue. 10b At least 2 DR ops are pending in the inbound ONION queue. 11b At least 3 DR ops are pending in the inbound ONION queue. • BIOS: 11b if !(F2x[1,0]90[BurstLength32] && (F2x[1,0]94[MemClkFreq]==000b)). • BIOS: 10b if (F2x[1,0]90[BurstLength32] && (F2x[1,0]94[MemClkFreq]==000b)).
2	**DisDramWrFlushForDR: disable DRAM write throttling for display refresh**. Read-write. Reset: 0. BIOS: 0. 1= Disable Fast Flushing of writes pending in the DCT while write throttling for DR is active.
1	**EnSameAddrSerBehindDROp: enable same address serialization behind display refresh request**. Read-write. Reset: 0. BIOS: 0. 1=A younger request that 39:6 address matches an older SeqId=0 and Coherent=0 link read request to DRAM waits at the IFQ CAM stage. 0=A younger request that 39:6 address matches an older SeqId=0 and Coherent=0 link read request to DRAM does not wait at the IFQ CAM stage.
0	**DisNbIntPendSbc**. Read-write. Reset: 0. BIOS: 0. 1=Disable NB from generating interrupt pending special bus cycle.

F3x190 Downcore Control Register

Cold reset: 0000 0000h.

Bits	Description
31:4	Reserved.
3:0	**DisCore[3:0]**. Read-only. 1=Disable the core. 0=Enable the core.

F3x1E4 SBI Control Register

See section 2.10.3 [Sideband Temperature Sensor Interface (SB-TSI)].

Bits	Description
31	**SbiRegWrDn: SBI register write done**. Reset 1b. Read-only. 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: 000b. Specifies bits[3:1] of the SMBus address of the processor SBI ports. SMBus address bits [7:1] is {1001b,~SbiAddr[2],SbiAddr[1:0]}.
3:2	Reserved.
1	**SbTsiDis: SMBus-based sideband interface thermal sensor disable**. Read-only. 1=The processor does not support SMBus-based SBI thermal sensor protocol.
0	Reserved.

F3x1E8 SBI Address Register

Reset 0000_0000h. The SB-TSI registers can be directly accessed by the processor using F3x1E8 and F3x1EC.

Access to these registers is accomplished as follows:

- Reads:
  - Write the register number to F3x1E8[SbiRegAddr].
  - Read the register contents from F3x1EC.

- Writes:
  - Write the register number to F3x1E8[SbiRegAddr].
  - Write all 32 bits to the register data to 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.

F3x1F0 Product Information Register

Bits	Description
31:19	Reserved.
18:16	**ClkDivC1E: clock divisor for C1E**. Read-only. Specifies the CpuDid value that must be programmed into the F3x[84:80][CpuDid] for C1E.
15:0	**BrandId**. Read-only. Brand identifier. This is identical to CPUID Fn8000_0001_EBX[BrandId].

F3x1FC Product Information Register

Bits	Description
31:0	Reserved.

3.7 Function 4 Link Configuration Registers

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

F4x00 Device/Vendor ID Register

Reset: 1304 1022h.

Bits	Description

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

F4x04 Status/Command Register

Reset: 0000 0000h. Read-only.

Bits	Description
31:16	**Status.**
15:0	**Command.**

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 0000h.

Bits	Description
31:8	Reserved.
7:0	**CapPtr: capabilities pointer.** Read-only, 00h. Specifies the offset of the link capabilities block based on whether the link supports unganged. 00h indicates that link 0 cannot be unganged.

F4x170 LMM Data Offset Register

Reset: 0000 0000h.

The link power management is configured by an array of registers called F4x174_x, which are defined following F4x174. [The LMM Data Offset Register] F4x170 and [The LMM Data Port] F4x174 are used to access F4x174_x. The register number (i.e., the number that follows "_x" in the register mnemonic) is specified by F4x170[LmmOffset]. Access to these registers is accomplished as follows:

- Reads:
  - Write the register number to F4x170[LmmOffset].
  - Read the register contents from F4x174.

- Writes:
  - Write the register number to F4x170[LmmOffset].
  - Write the register contents to F4x174.

Bits	Description
31:8	Reserved.
7:0	**LmmOffset: LMM controller offset**. Read-write.

F4x174 LMM Data Port

See F4x170 for details about this port.

F4x174_x[0F:00] LMM Configuration Registers

Cold reset (for all registers): 0000 0000h. Read-write.
The LMM Configuration Registers provide override controls for various register settings that affect power consumption. A particular set of overrides is selected based on LMAF[3:0] of the link management system management command. An LMAF of 0 indicates that no overrides are to be applied. A nonzero LMAF indicates that the programmed values are to be used to override the normal register values at the next LDTSTOP disconnect regardless of the setting of F0x16C[ImmUpdate], except as noted below for certain fields. 7 nonzero LMAF's are supported, corresponding to LMM Configuration Registers 1 to 7. A nonzero LMAF greater than 7 results in undefined behavior. See Table 42: [LmmOffset[3:0] Definition for the LMM Configuration Registers]. The format of the LMM configuration register is defined by Table 43: [LMM Configuration Register]. See "Link Power State Transitions" on page 36.

Table 42: LmmOffset[3:0] Definition for the LMM Configuration Registers

LmmOffset[3:0]	Definition
00h	LMM Configuration Register 0. Read-only.
07h-01h	LMM Configuration Registers 1 to 7. Read-write.
0Fh-08h	Reserved.

Table 43: LMM Configuration Register

Bits	Definition
31:28	Reserved.
27:26	**LmmTxLSSel[1:0]**. Defines behavior of lanes below the width programmed by F0x84[WidthOut]. Same definition as F0x170[TxLSSel].
25:24	**LmmRxLSSel[1:0]**. Defines behavior of lanes below the width programmed by F0x84[WidthOut]. Same definition as F0x170[RxLSSel].

Table 43: LMM Configuration Register

Bits	Definition
23:21	**LmmDeemph[2:0]: deemphasis**. Overrides the following fields of the Link Phy Deemphasis registers: F4x184_x[D4,C4][DCV], F4x184_x[D5,C5][PostCur1En, PostCur2En, PreCur1En, DL1, DL2, MapPostCur2En, MapPreCurEn, and DP1]. The values to be written to each of these fields is determined based on the deemphasis setting provided in this field, using the table in the description of the Link Phy Deemphasis registers. 000b: No deemphasis. 001b: -3 dB postcursor 010b: -6 dB postcursor 011b: -8 dB postcursor 100-111b: Reserved.
20:18	**LmmCpuDid[2:0]: clock divisor ID**. Same definition as F3x[84:80][CpuDid]. Affects any core in the C1 state. This field has no function if F3xD4[Smaf7Dis]=0. Takes effect immediately without LDTSTOP assertion.
17	**LmmCpuAltVidEn: alternate VID enable**. Same definition as F3x[84:80][CpuAltVidEn]. Affects all cores only if all cores are in the C1 state. This field has no function if F3xD4[Smaf7Dis]=0. Takes effect only with LDTSTOP assertion.
16	**LmmDramMemClkTri: DRAM memory clock tristated**. Same definition as F3x[84:80][DramMemClkTri]. Affects all cores only if all cores are in the C1 state. This field has no function if F3xD4[Smaf7Dis]=0 and also requires LmmCpuDid>0. Takes effect only with LDTSTOP assertion.
15	**LmmDramSr: DRAM self-refresh**. DRAM is placed into self-refresh. Same definition as F3x[84:80][DramSr]. Affects all cores only if all cores are in the C1 state. This field has no function if F3xD4[Smaf7Dis]=0 and also requires LmmCpuDid>0. Takes effect only with LDTSTOP assertion.
14	• **LmmCpuPrbEn**. Same definition as F3x[84:80][CpuPrbEn]. This field has no function if F3xD4[Smaf7Dis]=0. Takes effect immediately without LDTSTOP assertion.
13	Reserved.
12:11	**LmmTxInLnSt[1:0]: transmit inactive lane state**.Defines behavior of lanes below the width programmed by F0x84[WidthOut] but above the current width set by CDLW. Same encoding as F0x170[TxInLnSt[1:0]].
10:9	**LmmRxInLnSt[1:0]: receive inactive lane state**. Defines behavior of lanes below the width programmed by F0x84[WidthIn] but above the current width set by CDLW. Same encoding as F0x170[RxInLnSt[1:0]].
8:6	**LmmForceFullT0: force full T0 training time**. Same definition as F0x16C[ForceFullT0]].
5:0	**LmmT0Time: training 0 time**. Same definition as F0x16C[T0Time]].

F4x180 Link Phy Offset Register

Cold reset: 8000 0000h.

The link includes an array of registers called F4x184_x[NN:00], which are defined following F4x184. These are used primarily to control link electrical parameters and to program the link BIST engine. [The Link Phy Offset Register] F4x180 and [The Link Phy Data Port] F4x184 are used to access these registers. The register number (i.e., the number that follows “_x” in the register mnemonic) is specified by F4x180[LinkPhyOffset].

F4x180 and F4x184 are not generic registers and are not defined to be accessed in a manner other than described below. i.e A read to F4x184 is not ensured to equal the value previously written to F4x184. Access to these registers is accomplished as follows:

- Reads:
  - Write the register number to F4x180[LinkPhyOffset] with F4x180[LinkPhyWrite]=0.
  - Poll F4x180[LinkPhyDone] until it is high.
  - Read the register contents from F4x184. The same value can be read multiple times from F4x184 until F4x180 is written.

- Writes:
  - Write all 32 bits of register data to F4x184 (individual byte writes are not supported).
  - Write the register number to F4x180[LinkPhyOffset] with F4x180[LinkPhyWrite]=1.
  - Poll F4x180[LinkPhyDone] until it is high to ensure that the contents of the write have been delivered to the phy.

The link also includes 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:

- F4x180[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 F4x180[{UpperLinkPhyOffset, LinkPhyOffset}]. For example, to access [The DLL Control and Test 3] F4x184_x[530A, 520A], F4x180[{UpperLinkPhyOffset, LinkPhyOffset}] must be programmed as 4’h530A or 4’h520A.

Note: Read or write accesses to undocumented or undefined register numbers can result in undefined behavior.

Bits	Description
31	**LinkPhyDone: link phy access done**. Read-only. 1=The access to one of the F4x184_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 F4x184_x[NN:00] registers. 1=Write one of the F4x184_x[NN:00] registers.
29	**DirectMapEn: direct map enable**. Read-write. 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.
15:9	**UpperLinkPhyOffset: upper link phy offset address bits**. Read-write.
8:0	**LinkPhyOffset: link phy offset**. Read-write.

F4x184 Link Phy Data Port

See F4x180 for details about this port.

F4x184_x[D0,C0] Link Phy Impedance Registers

See F4x180 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 us 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, F4x184_xE0[RttRawCal]. 001b Rtt is as specified by the index field (F4x184_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. 011b Rtt is as specified by the sum: RttRawCal + RttIndex. If this results in a value that is greater than 1Fh, then 1Fh is used. 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] for more information about compensation.
28:21	Reserved.
20:16	**RttIndex: receiver termination resistance (Rtt) index**. Read-write. Cold reset: X. 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. 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] for more information about compensation. Bits Definition 000b Ron is as determined by the compensation circuit, F4x184_xE0[RonRawCal]. 001b Ron is as specified by the index field (F4x184_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. 011b Ron is as specified by the sum: RonRawCal + RonIndex. If this results in a value that is greater than 1Fh, then 1Fh is used. 100b Enable only one tap of the Ron resistor, as specified by RonIndex, and disable the base resistor that is normally always enabled. This is intended for testing purposes only. 101b-111b Reserved.
12:5	Reserved.
4:0	**RonIndex: transmitter resistance (Ron) index**. Read-write. Cold reset: X. See RonCtl for details about how this field is used.

F4x184_x[D1,C1] Link Phy Receiver Loop Filter Registers

See F4x180 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].

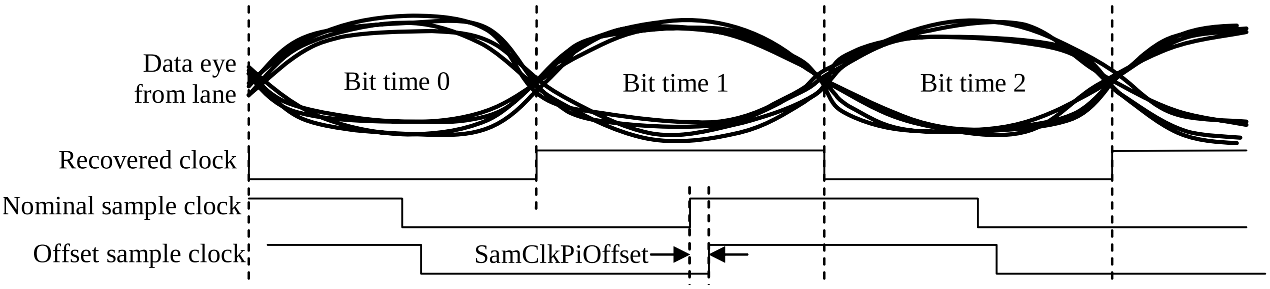

Figure 11: Link phy recovered clock and sample clock

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 upward or downward; if it trends downward, it hits zero and then trend 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. The BIOS should program this field to 20h if link frequency is 1.2GHz or greater or 10h if link frequency is less than 1.2GHz.

21:14	**LfcMin: loop filter counter minimum value**. Read-write. Cold reset: 40h. The BIOS should program this field to 10h if link frequency is 1.2GHz or greater or 8h if link frequency is less than 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 11. 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 11.
2:0	Reserved.

F4x184_x[D4,C4] Link Phy DFR Control Registers

See F4x180 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 _xC4 and _xD4 registers must be configured identically and must not be modified after link power management is enabled, or undefined behavior may result. These registers always returns the active setting, regardless of if there is a change pending from a software write or an LMM override (as configured by F4x174_x[0F:00]).

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
-2dB postcursor	19h
-3dB postcursor	16h
-6dB postcursor	10h
-8dB postcursor	0Dh
-9dB postcursor	0Bh

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.
9:0	Reserved.

F4x184_x[D5,C5] Link Phy Deemphasis Value Registers

See F4x180 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] for more information about deemphasis. These register bits are updated as specified by F0x16C[ImmUpdate]. The _xC5 and _xD5 registers must be configured identically and must not be modified after link power management is enabled, or undefined behavior may result. Reads of these registers always return the active setting, regardless of if there is a change pending from a software write or an LMM override (as configured by F4x174_x[0F:00]).

In Gen3 link DC-coupled mode, three postcursor deemphasis settings, -3dB, -6dB and -8dB are supported. In addition, an -11dB postcursor deemphasis setting and an -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.

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. IF (Gen3) THEN BIOS: 1. ELSE BIOS: 0. ENDIF. 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. BIOS: 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. BIOS: 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	Reserved.

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 - VLDT * 0.5 * (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. IF (Gen3) THEN BIOS: 12h. ELSE BIOS: 00h. ENDIF.
15:13	Reserved.
12:8	**DL2: deemphasis level 2**. Read-write. Cold reset: 0. BIOS: 0.
7	**TxLS23ClkGateEn: Enable shutting off internal clocks to Tx lanes in LS2 or PHY OFF**. Read-write; changes take effect on next warm reset or LDTSTOP disconnect. Cold reset: 0b. IF (Gen3) THEN BIOS: 1. ELSE BIOS: 0. ENDIF. 1=Enable shutting off internal clocks to Tx lanes in LS2 or PHY OFF to save power.
6	**MapPostCur2En: Map post-cursor 2 deemphasis enable**. Read-write. Cold reset: 0. BIOS: 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. BIOS: 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. BIOS: 0.

F4x184_x[DF,CF] Link FIFO Read Pointer Optimization Registers

Cold reset: 0000 0000h.

See F4x180 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 processor NB clock domain and the link clock domain. At cold reset, the read pointer and write pointer for the TX FIFO is positioned conservatively (32 bit-times apart), such that FIFO latency may be greater than is necessary. F4x184_x[DF,CF][XmtRdPtr] may be used to position the read pointer and write pointer of the TX FIFO closer to each other such that latency is reduced. After writing to this register, the new values are applied to the TX FIFO at the next LDTSTOP disconnect or warm reset. Reads from the register always return the pending value from the last write.

Bits	Description
31:8	Reserved.
7:4	**XmtRdPtr: transmit FIFO read pointer**. Read-write; changes take effect on next link disconnect. 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	Reserved.

F4x184_xE0 Link Phy Compensation Control Register

See F4x180 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
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.

F4x184_x100 Link BIST Control Register

See F4x180 for register access information.

Bits	Description
31	**Width**. Read-only. Cold reset: 0. Indicates the implemented width of the BIST engine. Bits — Status 0 — 8 bits. 1 — 16 bits. The same patterns are transmitted on the upper and lower sublinks. The upper bit of F0x170[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.
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] F4x184_x110 and [The Link BIST Northbound RX Inversion Register] F4x184_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. A 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.

F4x184_x104 Link BIST Southbound TX Pattern Control Register

See F4x180 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 F4x184_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

F4x184_x108 Link BIST Southbound TX Pattern Buffer 1 Register

See F4x180 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.

F4x184_x10C Link BIST Southbound TX Mask Register

See F4x180 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

F4x184_x110 Link BIST Southbound TX Inversion Register

See F4x180 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 F4x184_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 width link and the initial pattern in the inversion register must repeat on 9-bit boundaries.

F4x184_x114 Link BIST Southbound TX Pattern Buffer 2 Register

See F4x180 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.

F4x184_x118 Link BIST Southbound TX Pattern Buffer 2 Enable Register

See F4x180 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

F4x184_x11C Link BIST Southbound TX Pattern Buffer Extension Register

See F4x180 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.

F4x184_x120 Link BIST Southbound TX Scramble Register

See F4x180 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

F4x184_x124 Link BIST Northbound RX Pattern Control Register

See F4x180 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 F4x184_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

F4x184_x128 Link BIST Northbound RX Pattern Buffer 1 Register

See F4x180 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.

F4x184_x12C Link BIST Northbound RX Mask Register

See F4x180 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 a 4-bit link and bits 7:2 for a 2-bit link. Bit Lane Bit Lane 0 CAD0 5 CAD5 1 CAD1 6 CAD6 2 CAD2 7 CAD7 3 CAD3 8 CTL 4 CAD4

F4x184_x130 Link BIST Northbound RX Inversion Register

See F4x180 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 F4x184_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 a 16/8-bit link and the initial pattern in the inversion register must repeat on 9-bit boundaries.

F4x184_x134 Link BIST Northbound RX Pattern Buffer 2 Register

See F4x180 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.

F4x184_x138 Link BIST Northbound RX Pattern Buffer 2 Enable Register

See F4x180 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

F4x184_x13C Link BIST Northbound RX Pattern Buffer Extension Register

See F4x180 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.

F4x184_x140 Link BIST Northbound RX Scramble Register

See F4x180 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

F4x184_x144 Link BIST Northbound RX Error Status Register

See F4x180 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, undefined operation may result); 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

F4x184_x[530A, 520A] DLL Control and Test 3

This is a direct map register, see F4x180 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. The value specified by Ls2ExitTime must be less than the value specified by F0x16C[T0Time], or it can cause undefined behavior. Bits Definition 000b Delay=10us. 001b Delay=5us. 010b Delay=2.5us. 011b Delay=1.25us. 100b Delay=625ns. 101b Delay=0s. 11xb Reserved.
28:18	Reserved.
17	**DllLockFastModeEn: DLL lock fast mode enable**. Read-write. Cold reset: 0. BIOS: 0. 1=Enables DLL lock fast mode. 0=DLL lock operates at standard speed.
16:15	Reserved.
14:13	**AnalogWaitTime: analog wait time to turn on DLL**. Read-write. Cold reset: 00b. BIOS: 00b. This field is used with DllAnalogOkIgnore; if DllAnalogOkIgnore is set, the turning on of DLL circuit after cold reset is delayed by a timer specified by this field, encoded as follows. Bits Definition 00b Delay=1.25us. 01b Delay=0.625us. 10b Delay=2.5us. 11b Delay=0.3125us.
12:11	Reserved.
10	**DllAnalogOkIgnore: DLL analog start signal ignore**. Read-write. Cold reset: 0. BIOS: 1. 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.
9:8	Reserved.
7	**BiasDisInLs2: bias disable in LS2 power state**. Read-write. Cold reset: 0. BIOS: 1. 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 LS0 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 LS0 power state transition speed. 1=Fast transition mode selected. 0=Slow transition mode selected.

3.8 APIC Registers

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

APIC20 APIC ID Register

Bits	Description
31:24	**ApicId[7:0]: APIC identification**. Read-write. Reset: {0000000b,CpuCoreNum[0]}. Software must ensure that all APICs are assigned unique ApicId values. When F0x68[ApicExtId,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 [Number of Cores and Core Number].
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 that 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 enters servicing. 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 44: 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**. 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.
11:8	Reserved.
7:0	**Vector**. Read-write. This field contains the vector that is sent for this interrupt source.

APIC330 Thermal Local Vector Table Entry

Reset: 0001 0000h. Interrupts for this local vector table are caused by changes in [The P-state Current Limit Register] MSRC001_0061[CurPstateLimit] due to HTC.

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

12	**DlvryStat: delivery status**. Read-only. This bit is set to indicate that the interrupt has not yet been accepted by the core.
11	Reserved.
10:8	**MsgType: message type**. Read-write. Only the following message types are legal: • 000b = Fixed • 010 = SMI • 100 = NMI
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. Only the following message types are legal: • 000b = Fixed • 010 = SMI • 100 = NMI
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 link messages instead of individual pins.
12	**DlvryStat: delivery status**. Read-only. This bit is set to indicate that the interrupt has not yet been accepted by the core.

11	Reserved.
10:8	**MsgType: message type**. Read-write. Only the following message types are legal: • 000b = Fixed • 010 = SMI • 100 = NMI
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
31:17	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 link messages instead of individual pins.
12	**DlvryStat: delivery status**. Read-only. This bit is set to indicate that the interrupt has not yet been accepted by the core.
11	Reserved.
10:8	**MsgType: message type**. Read-write. Only the following message types are legal: • 000b = Fixed • 010 = SMI • 100 = NMI
7:0	**Vector**. Read-write. This field contains the vector that is sent for this interrupt source.

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**. Read-write. Only the following message types are legal: • 000b = Fixed • 010 = SMI • 100 = NMI
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:

Table 45: Div Bit Encoding

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.

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: SEOI enable**. Read-write. 1=Enable SEOI generation when a write to the SEOI register is received. This bit must be set before any writes are generated to the SEOI register.
0	**IerEn: IER enable**. Read-write. 1=Enable writes to the IER 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.

There are no internal interrupt conditions that use these interrupts.

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. Supported message types are: 000b (Fixed), 010b (SMI), 100b (NMI), and 111b (ExtINT); all others are reserved.
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] also.

Unless otherwise specified, the 1-bit feature fields are encoded as 1=Feature is supported by the processor; 0=Feature is not supported by the processor.

The following provides AMD Family 11h specific details about CPUID. See the *CPUID Specification* for further information.

CPUID Fn0000_0000 Processor Vendor and Largest Standard Function Number

Register	Bits	Description
EAX	31:0	The largest CPUID standard-function input value supported by the processor implementation: 0000_0001h.
EBX, ECX, EDX	31:0	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 Fn0000_0001_EAX Family, Model, Stepping Identifiers

This register provides identical information to CPUID Fn8000_0001_EAX and 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]=02h, then Family[7:0]=11h. This document applies only to family 11h 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**: 02h.
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. Changes to APIC20[ApicId] do not affect the value of this CPUID register. See also section 2.9.2 [Number of Cores and Core Number].
23:16	**LogicalProcessorCount**: If CPUID Fn0000_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 Fn0000_0001_ECX Feature Identifiers

Bits	Description
31:24	Reserved.

23	**POPCNT**: POPCNT instruction = 0.
22:14	Reserved.
13	**CMPXCHG16B**: CMPXCHG16B instruction = 1.
12:4	Reserved.
3	**Monitor**: Monitor/Mwait instructions = 0.
2:1	Reserved.
0	**SSE3**: SSE3 extensions = 1; may be overridden by MSRC001_0015[SseDis].

CPUID Fn0000_0001_EDX Feature Identifiers

Bits	Description
31:29	Reserved.
28	**HTT**: hyper-threading technology. This bit qualifies the meaning of CPUID Fn0000_0001_EBX[LogicalProcessorCount]. (setting varies by product). 1=Dual core product. 0=Single core product.
27	Reserved.
26	**SSE2**: SSE2 extensions = 1; may be overridden by MSRC001_0015[SseDis].
25	**SSE**: SSE extensions = 1; may be overridden by MSRC001_0015[SseDis].
24	**FXSR**: FXSAVE and FXRSTOR instructions = 1.
23	**MMX**: MMX™ instructions = 1.
22:20	Reserved.
19	**CLFSH**: CLFLUSH instruction = 1.
18	Reserved.
17	**PSE36**: page-size extensions = 1.
16	**PAT**: page attribute table = 1.
15	**CMOV**: conditional move instructions, CMOV, FCOMI, FCMOV = 1.
14	**MCA**: machine check architecture, MCG_CAP = 1.
13	**PGE**: page global extension, CR4.PGE = 1.
12	**MTRR**: memory-type range registers = 1.
11	**SysEnterSysExit**: SYSENTER and SYSEXIT instructions = 1.
10	Reserved.
9	**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	**CMPXCHG8B**: CMPXCHG8B instruction = 1.
7	**MCE**: machine check exception, CR4.MCE = 1.
6	**PAE**: physical-address extensions (PAE) = 1.
5	**MSR**: AMD model-specific registers (MSRs), with RDMSR and WRMSR instructions = 1.
4	**TSC**: time stamp counter, RDTSC/RDTSCP instructions, CR4.TSD = 1.
3	**PSE**: page-size extensions (4 MB pages) = 1.
2	**DE**: debugging extensions, IO breakpoints, CR4.DE = 1.

1	**VME**: virtual-mode enhancements = 1.
0	**FPU**: x87 floating point unit on-chip = 1.

CPUID Fn8000_0000 Processor Vendor and Largest Extended Function Number

Register	Bits	Description
EAX	31:0	The largest CPUID extended-function input value supported by the processor implementation: 8000_001Ah.
EBX, ECX, EDX	31:0	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 Fn8000_0001_EAX AMD Family, Model, Stepping

Same as CPUID Fn0000_0001_EAX.

CPUID Fn8000_0001_EBX BrandId Identifier

Bits	Description
31:28	**PkgType**: package type = 2h. Specifies the S1g2 package.
27:16	Reserved.
15:0	**BrandId**: brand ID. This is identical to F3x1F0[BrandId].

CPUID Fn8000_0001_ECX Feature Identifiers

Bits	Description
31:14	Reserved.
13	**WDT**: Watchdog timer support = 0.
12	**SKINIT**: SKINIT and STGI support = 1.
11	**SSE5** instruction support = 0
10	**IBS**: Instruction Based Sampling = 0.
9	**OSVW**: OS Visible Work-around support = 1.
8	**3DNowPrefetch**: Prefetch and PrefetchW instructions = 1.
7	**MisAlignSse**: Misaligned SSE Mode = 0.
6	**SSE4A**: EXTRQ, INSERTQ, MOVNTSS, and MOVNTSD instruction support = 0.
5	**ABM**: Advanced bit manipulation. LZCNT instruction support = 0.
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). Indicates support for: VMRUN, VMLOAD, VMSAVE, CLGI, VMMCALL, and INVLPGA.

1	**CmpLegacy: core multi-processing legacy mode**. Setting varies by product. 1=Dual core product (CPUID Fn8000_0008[NC] != 0). 0=Single core product (CPUID Fn8000_0008[NC] == 0).
0	**LahfSahf: LAHF/SAHF instructions**. Reset: 1.

CPUID Fn8000_0001_EDX Feature Identifiers

Bits	Description
31	**3DNow**: 3DNow!™ instructions = 1.
30	**3DNowExt**: AMD extensions to 3DNow!™ instructions = 1.
29	**LM**: long mode (may vary by product).
28	Reserved.
27	**RDTSCP**: RDTSCP instruction = 1.
26	**Page1GB**: 1 GB large page support = 0.
25	**FFXSR**: FXSAVE and FXRSTOR instruction optimizations = 1.
24	**FXSR**: FXSAVE and FXRSTOR instructions = 1.
23	**MMX**: MMX™ instructions = 1.
22	**MmxExt**: AMD extensions to MMX™ instructions = 1.
21	Reserved.
20	**NX**: no-execute page protection = 1.
19:18	Reserved.
17	**PSE36**: page-size extensions = 1.
16	**PAT**: page attribute table = 1.
15	**CMOV**: conditional move instructions, CMOV, FCOMI, FCMOV = 1.
14	**MCA**: machine check architecture, MCG_CAP = 1.
13	**PGE**: page global extension, CR4.PGE = 1.
12	**MTRR**: memory-type range registers = 1.
11	**SysCallSysRet**: SYSCALL and SYSRET instructions = 1.
10	Reserved.
9	**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	**CMPXCHG8B**: CMPXCHG8B instruction = 1.
7	**MCE**: machine check exception, CR4.MCE = 1.
6	**PAE**: physical-address extensions (PAE) = 1.
5	**MSR**: AMD model-specific registers (MSRs), with RDMSR and WRMSR instructions = 1.
4	**TSC**: time stamp counter, RDTSC/RDTSCP instructions, CR4.TSD = 1.
3	**PSE**: page-size extensions (4 MB pages) = 1.
2	**DE**: debugging extensions, IO breakpoints, CR4.DE = 1.
1	**VME**: virtual-mode enhancements = 1.
0	**FPU**: x87 floating point unit on-chip = 1.

CPUID Fn8000_000[4:2] 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 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 = 8. The value returned is for the number of entries available for the 2 MB page size; 4 MB pages require two 2 MB entries, so the number of entries available for the 4 MB page size is one-half the returned value.
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 = 8. The value returned is for the number of entries available for the 2 MB page size; 4 MB pages require two 2 MB entries, so the number of entries available for the 4 MB page size is one-half the returned value.
EBX	31:24	Data TLB associativity for 4 KB pages = FFh.
EBX	23:16	Data TLB number of entries for 4 KB pages = 32.
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.	8h: 16-way associative.
1h: Direct mapped.	Ah: 32-way associative.
2h: 2-way associative.	Bh: 48-way associative.
4h: 4-way associative.	Ch: 64-way associative.
6h: 8-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 = 0.
EAX	27:16	**L2DTlb2and4MSize**. L2 data TLB number of entries for 2 MB and 4 MB pages = 0. The value returned is for the number of entries available for the 2 MB page size; 4 MB pages require two 2 MB entries, so the number of entries available for the 4 MB page size is one-half the returned value.
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	**L2Asssoc**. 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	31:16	**L3Size**. L3 cache size = 0000h (No L3 cache).
EDX	15:12	**L3Assoc**. L3 cache associativity = 0.
EDX	11:8	**L3LinesPerTag**. L3 cache lines per tag = 0.
EDX	7:0	**L3LineSize**. L3 cache line size in bytes = 0.

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** = 1: The TSC rate is ensured to be invariant across all P-states, C-States, and stop-grant transitions (such as STPCLK Throttling).
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.

Register	Bits	Description
EDX	6	**100MHzSteps**: 100 MHz multiplier Control = 1.
EDX	5	Reserved.
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 Fn8000_0001_EDX[LM]) then this is 30h; else this is 20h.
EAX	7:0	Maximum physical byte address size in bits = 28h.
EBX	31:0	Reserved.
ECX	31:16	Reserved.
ECX	15:12	**ApicIdCoreIdSize[3:0]**. The number of least significant bits in the Initial APIC ID that indicate core ID within a processor = 1h.
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). See also section 2.9.2 [Number of Cores and Core Number].
EDX	31:0	Reserved.

CPUID Fn8000_0009 Reserved

CPUID Fn8000_000A_EAX SVM Revision and Feature Identification

This provides SVM revision and feature information. If CPUID Fn8000_0001_ECX[SVM]=0 then CPUID Fn8000_000A_EAX is reserved.

Bits	Description
31:8	Reserved.
7:0	**SvmRev: SVM revision.** Reset: 01h.

CPUID Fn8000_000A_EBX SVM Revision and Feature Identification

This provides SVM revision and feature information. If CPUID Fn8000_0001_ECX[SVM]=0 then CPUID Fn8000_000A_EBX is reserved.

Bits	Description
31:0	**NASID: number of address space identifiers (ASID).** Reset: 40h.

CPUID Fn8000_000A_ECX SVM Revision and Feature Identification

This provides SVM revision and feature information. If CPUID Fn8000_0001_ECX[SVM]=0 then CPUID Fn8000_000A_ECX is reserved.

Bits	Description
31:0	Reserved.

CPUID Fn8000_000A_EDX SVM Revision and Feature Identification

This provides SVM revision and feature information. If CPUID Fn8000_0001_ECX[SVM]=0 then CPUID Fn8000_000A_EDX is reserved.

Bits	Description
31:4	Reserved.
3	**NRIPS: NRIP save.** Reset: 1. Indicates support for NRIP save on #VMEXIT.
2	**SVML: SVM lock.** Reset: 1.
1	**LbrVirt: LBR virtualization.** Reset: 1.
0	**NP: nested paging.** Reset: 0.

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 = 0.
EAX	27:16	L1 data TLB associativity for 1 GB pages = 0.
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** = 0.
EAX	0	**FP128** = 0.
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] for a description of the register naming convention. MSRs are accessed through x86 WRMSR and RDMSR instructions.

MSR0000_0000 Load-Store MCA Address Register

MSR0000_0000 alias of MSR0000_040E.

MSR0000_0001 Load-Store MCA Status Register

MSR0000_0001 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
63:40	MBZ.
39:12	**ApicBar: APIC base address register.** Read-write. Reset: 00_FEE0_0h. Specifies the base address for the APICXX register set. See section 3.8 [APIC Registers] 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 BSC. 0=The core is not the BSC.
7:0	MBZ.

MSR0000_002A Cluster ID Register (EBL_CR_POWERON)

Reset: 0000 0000 0000 0000h. Attempted writes to this register result in general protection faults with error code 0.

Bits	Description
63:18	RAZ.
17:16	**ClusterID: APIC cluster ID.** Read-only. This is normally 00b; the value does not affect hardware.
15:0	RAZ.

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. Read-write.
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	RAZ.
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 0105h.

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

MSR0000_017A Global Machine Check Status Register (MCG_STAT)

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

Bits	Description
63:3	Reserved.
2	**MCIP: machine check in progress**. Read-write; set-by-hardware. 1=A machine check is in progress.
1	**EIPV: error instruction pointer valid**. Read-write; set or cleared 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; set or cleared 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].

Bits	Description
63:5	Reserved
4	**NBE: NB register bank enable**. Read-write. 1=The NB 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. For more information about this MSR, see the APM2 section titled “Debug-Control MSR (DebugCtlMSR)”.

Bits	Description
63:7	Reserved.
6	MBZ.
5:2	**PB: performance monitor pin control**. Read-write. 1=PB[i] reports breakpoint information. 0=PB[i] reports performance-monitor information.
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)

For more information about this MSR, see the APM2 section titled “Control-Transfer Recording MSRs”.

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)

For more information about this MSR, see the APM2 section titled “Control-Transfer Recording MSRs”.

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

For more information about this MSR, see the APM2 section titled “Control-Transfer Recording MSRs”.

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

MSR0000_01DE Last Exception To IP Register

For more information about this MSR, see the APM2 section titled “Control-Transfer Recording MSRs”.

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]. 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]. This bit can be enabled for fixed MTRR ranges only (see MSRC001_0010[MtrrFixDramEn, MtrrFixDramModEn]); 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[39:12] & PhyMask[39:12] == PhyBase[39:12] & PhyMask[39:12].

For example, if the variable MTRR spans 256K bytes and starts at the 1M byte address. The PhyBase would be set to 00_0010_0000h and the PhyMask to FF_FFFC_0000h (with zeros filling in for bits[11:0]). This results in a range from 00_0010_0000h to 00_0013_FFFFh.

MSR0000_020[E, C, A, 8, 6, 4, 2, 0] (MTRRphysBase*n*)

Bits	Description
63:40	MBZ.
39:12	**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
63:40	MBZ.
39:12	**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 1_7000 to 1_7FFF (plus the base).
55:48	**MemType: memory type**. Read-write. Address range from 1_6000 to 1_6FFF (plus the base).
47:40	**MemType: memory type**. Read-write. Address range from 1_5000 to 1_5FFF (plus the base).
39:32	**MemType: memory type**. Read-write. Address range from 1_4000 to 1_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.	5h = WP or write protect.
1h = WC or write combining.	6h = WB or write back.
4h = WT or write through.	7h = UC- or uncacheable (overridden by MTRR WC state)
All other values result in a #GP(0).	

Bits	Description
63:59	MBZ.
58:56	**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. The memory type for memory space that is not specified by either the fixed or variable range MTRR's is defined as a function of MtrrDefTypeEn as follows: • If MtrrDefTypeEn==1 then the default memory type is MemType. • If MtrrDefTypeEn==0 then the default memory type is UC.

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]. 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.
1	**ECCM: multi-bit ECC data errors**. Report multi-bit ECC data errors during data 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 data cache line fills or TLB reloads from the internal L2 or the system.

MSR0000_0401 DC Machine Check Status Register (MC0_STATUS)

Cold Reset: xxxx_xxxx_xxxx_xxxxh.
See section 2.13.1 [Machine Check Architecture]. Each of the MCi_STATUS registers hold information identifying the last error logged in each bank. Software is normally only allowed to write 0's to these registers to clear the fields so subsequent errors may be logged. See 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].
61	**UC: error uncorrected**. Read-write; set or cleared by hardware. 1=The error was not corrected by hardware.
60	**EN: error enable**. Read-write; set or cleared by hardware. 1=MCA error reporting is enabled for this error in MCi_CTL.
59	**MISCV: miscellaneous error register valid**. Read-only. 1=MCi_MISC contains valid information for this error. This bit is always 0, except in the case of [The Reserved] MSR0000_0413.
58	**ADDRV: error address valid**. Read-write; set or cleared by hardware. 1=The address saved in MCi_ADDR is the address where the error occurred.

57	**PCC: processor context corrupt**. Read-write; set or cleared 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 and ECC error is detected. See Table 48 for the mapping that shows which bit errors result in which syndrome values for 8-bit ECC. • MC[3:1]_STATUS (LS, BU, IC): Reserved.
46	**CECC: Correctable ECC Error**. Read-write; set or cleared by hardware. 1=The error was a correctable ECC error.
45	**UECC: Uncorrectable ECC Error**. Read-write; set or cleared by hardware. 1=The error was an uncorrectable ECC error.
44:41	Reserved.
40	**SCRUB: error detected on a scrub**. Read-write; set or cleared by hardware. • MC0_STATUS (DC): 1=The error was detected on a scrub. • MC[3:1]_STATUS (LS, BU, IC): Reserved.
39:20	Reserved.
19:16	**ErrorCodeExt: extended error code**. Read-write. The following extended error codes are defined for MC0_STATUS (DC) and MC1_STATUS (IC): 0000b = TLB parity error in physical array. 0001b = TLB parity error in virtual array (multi-match error). The following extended error codes are defined for MC2_STATUS (BU): 0000b = Bus or cache data array error. 0010b = Cache tag array error. MC3_STATUS (LS): Reserved
15:0	**ErrorCode: error code**. Read-write. See error-code tables below.

This register reports these DC errors:

Table 46: 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 46: 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 47: DC error signatures

Error Type	[19:16] Error-Code-Ext	Error Code (see F3x48 for encoding)						[61] UC	[58] ADD-RV	[57] PCC	[54:47] Synd Valid	[46] CECC	[45] UECC	[40] SCR UB
		Type	10:9 PP	8 T	7:4 RRRR	3:2 II/TT	1:0 LL							
System Line Fill	0000	Bus	SRC	0	DRD	MEM /IO	LG	If multi-bit	1	If multi-bit	Y	If single-bit	If multi-bit	0
L2 Cache Line Fill	0000	Mem	-	-	DRD	Data	L2	If multi-bit	1	If multi-bit	Y	If single-bit	If multi-bit	0
Data Load/ Store/ Victim/ Snoop	0000	Mem	-	-	DRD/ DWR/ Evict/ Snoop	Data	L1	1	1/0	1	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 Multi-match	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 Multi-match	0001	TLB	-	-	-	Data	L2	1	1	1	N	0	0	0

The Syndrome field uniquely identify the position of a single-bit ECC error. The following table lists the syndrome for a single bit error in either the 64 data bits or 8 check bits.

Table 48: 8-bit ECC Syndromes

	n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7
Data (0+n)	CEh	CBh	D3h	D5h	D6h	D9h	DAh	DCh
Data (8+n)	23h	25h	26h	29h	2ah	2ch	31h	34h
Data (16+n)	0Eh	0Bh	13h	15h	16h	19h	1Ah	1Ch
Data (24+n)	E3h	E5h	E6h	E9h	EAh	ECh	F1h	F4h
Data (32+n)	4Fh	4Ah	52h	54h	57h	58h	5Bh	5Dh
Data (40+n)	A2h	A4h	A7h	A8h	ABh	ADh	B0h	B5h
Data (48+n)	8Fh	8Ah	92h	94h	97h	98h	9Bh	9Dh

Table 48: 8-bit ECC Syndromes

	n=0	n=1	n=2	n=3	n=4	n=5	n=6	n=7
Data (56+n)	62h	64h	67h	68h	6Bh	6Dh	70h	75h
Check (n)	01h	02h	04h	08h	10h	20h	40h	80h

MSR0000_0402 DC Machine Check Address Register (MC0_ADDR)

Cold Reset: xxxx_xxxx_xxxx_xxxxh.

See section 2.13.1 [Machine Check Architecture]. Each of the MCi_ADDR registers are written to by hardware and read-write accessible by software. MCi_ADDR registers contains valid data if indicated by MCi_STATUS[ADDRV]. Table 49 defines the address register as a function of error type.

Table 49: DC error data; address register

Error Type	Memory Transaction Type (RRRR; Table 29)	Address Register Bits	Description
System Line Fill	DRD	39:6	Physical address
L2 Cache Line Fill			
Data Load/ Store/ Victim/ Snoop	DRD	39:4[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 14:3.
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]. 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[S_RDE_ALL] = 1.
8:7	Reserved.
6	**L2TP: L2 TLB parity errors**. Report instruction cache L2 TLB parity errors.
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]. 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 50: IC error signatures

Error Type	**[19:16] Error-Code-Ext**	**Error Code (see F3x48 for encoding)**						**[61] UC**	**[58] ADD-RV**	**[57] PCC**	**[54:47] Synd Valid**	**[46] CECC**	**[45] UECC**	**[40] SCRUB**
		Type	**10:9 PP**	**8 T**	**7:4 RRRR**	**3:2 II/TT**	**1:0 LL**							
System Data Read Error	0000	BUS	SRC	0	IRD	MEM	LG	1	0	0	N	0	0	0
L2 Cache Line Fill	0000	Memory	-	-	IRD	Instr	L2	0[1]	1	0	N	0[2]	1	0
IC Data Load (Parity)	0000	Memory	-	-	IRD	Instr	L1	0	1	0	N	0	0	0
Tag Snoop	0000	Memory	-	-	Snoop	Instr	L1	1	1	1	N	0	0	0
Copyback parity	0000	Memory	-	-	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 Multi-match	0001	TLB	-	-	-	Instr	L1	0	1	0	N	0	0	0

Table 50: IC error signatures

Error Type	[19:16] Error-Code-Ext	Error Code (see F3x48 for encoding)						[61] UC	[58] ADDRV	[57] PCC	[54:47] Synd Valid	[46] CECC	[45] UECC	[40] SCRUB
		Type	10:9 PP	8 T	7:4 RRRR	3:2 II/TT	1:0 LL							
L2 TLB	0000	TLB	-	-	-	Instr	L2	0	1	0	N	0	0	0
L2 TLB Multi-match	0001	TLB	-	-	-	Instr	L2	0	1	0	N	0		0

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

MSR0000_0406 IC Machine Check Address Register (MC1_ADDR)

Cold Reset: xxxx_xxxx_xxxx_xxxxh. See section 2.13.1 [Machine Check Architecture]. Each of the MCi_ADDR registers are written to by hardware and read-write accessible by software. MCi_ADDR registers contains valid data if indicated by MCi_STATUS[ADDRV]. Table 51 defines the address register as a function of error type.

Table 51: IC error data; address register

Error Type	Address Register Bits	Description
L2 Cache Line Fill	39:6	Physical address
IC Data Load	47:4	Linear address
Tag Snoop	39:6	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]. For all bits, 1=Enable the specified reporting mechanism.

Bits	Enable
63:20	Reserved.
19	**L2T_ECCM_SCR: L2 tag array multi-bit ECC scrub**. Report L2 tag array multi-bit ECC errors for a scrub.
18	**L2T_ECC1_SCR: L2 tag array 1-bit ECC scrub**. Report L2 tag array 1-bit ECC errors for a scrub.

Bits	Enable
17	**L2D_ECCM_CPB: L2 data array multi-bit ECC copyback**. Report L2 data array multi-bit ECC errors for a copyback.
16	**L2D_ECCM_SNP: L2 data array multi-bit ECC snoop**. Report L2 data array multi-bit ECC errors for a snoop.
15	**L2D_ECCM_TLB: L2 data array multi-bit ECC TLB reload**. Report L2 data array multi-bit ECC errors for a TLB reload.
14	**L2D_ECC1_CPB: L2 data array 1-bit ECC copyback**. Report L2 data array 1-bit ECC errors for a copyback.
13	**L2D_ECC1_SNP: L2 data array 1-bit ECC snoop**. Report L2 data array 1-bit ECC errors for a snoop.
12	**L2D_ECC1_TLB: L2 data array 1-bit ECC TLB reload**. Report L2 data array 1-bit ECC errors for a TLB reload.
11	**L2T_PAR_SCR: L2 tag array parity scrub**. Report L2 tag array parity errors for a scrub.
10	**L2T_PAR_CPB: L2 tag array parity copyback**. Report L2 tag array parity errors for a copyback.
9	**L2T_PAR_SNP: L2 tag array parity snoop**. Report L2 tag array parity errors for a snoop.
8	**L2T_PAR_TLB: L2 tag array parity TLB reload**. Report L2 tag array parity errors for a TLB reload.
7	**L2T_PAR_ICDC: L2 tag array parity IC or DC fetch**. Report L2 tag array parity errors for an IC or DC fetch.
6	**S_ECCM_HP: system data multi-bit ECC hardware prefetch**. Report system data multi-bit ECC errors for a hardware prefetch.
5	**S_ECCM_TLB: system data multi-bit ECC TLB reload**. Report system data multi-bit ECC errors for a TLB reload.
4	**S_ECC1_HP: system data 1-bit ECC hardware prefetch**. Report system data 1-bit ECC errors for a hardware prefetch.
3	**S_ECC1_TLB: system data 1-bit ECC TLB Reload**. Report system data 1-bit ECC errors for a TLB reload.
2	**S_RDE_ALL: all system read data**. Report system read data errors for any operation including a DC/IC fetch, TLB reload or hardware prefetch.
1	**S_RDE_TLB: system read data TLB reload**. Report system read data errors for a TLB reload.
0	**S_RDE_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]. 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 52: BU error signatures

Error Type	Access Type	[19:16] Error-Code-Ext	Error Code (see F3x48 for encoding)						[61] UC	[58] ADD-RV	[57] PCC	[54:47] Synd Valid	[46] CECC	[45] UECC	[40] SCRUB
			Type	10:9 PP	8 T	7:4 RRRR	3:2 II/TT	1:0 LL							
Tag Parity	Instr Fetch	0010	Mem	-	-	IRD	Instr	L2	1	1	1	N	0	0	0
	Data Fetch	0010	Mem	-	-	DRD	Data	L2	1	1	1	N	0	0	0
	TLB/ Snoop/ Evict	0010	Mem	-	-	RD/ Snoop/ Evict	Gen	L2	1	1	1	N	0	0	0
	Scrub	0010	Mem	-	-	GEN	Gen	L2	1	1	0	N	0	0	1
Tag ECC	Scrub	0010	Mem	-	-	GEN	Instr	L2	If multi-bit	1	0	N	If single-bit	If multi-bit	1
System Data Read Error	TLB	0000	BUS	SRC	0	RD	MEM	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
System Data ECC	TLB	0000	BUS	SRC	0	RD	MEM /IO	LG	If multi-bit	1	If multi-bit	N	If single-bit	If multi-bit	0
	HW Prefetch	0000	BUS	SRC	0	Prefetch	MEM	LG	If multi-bit	0	If multi-bit	N	If single-bit	If multi-bit	0
Data ECC	TLB	0000	Mem	-	-	RD	Gen	L2	If multi-bit	0	If multi-bit	N	If single-bit	If multi-bit	0
Data Copy-back	Snoop/ Evict	0000	Mem	-	-	RD	Gen	L2	If multi-bit	0	If multi-bit	N	If single-bit	If multi-bit	0

MSR0000_040A BU Machine Check Address Register (MC2_ADDR)

Cold Reset: xxxx_xxxx_xxxx_xxxxh. See section 2.13.1 [Machine Check Architecture]. Each of the MCi_ADDR registers are written to by hardware and read-write accessible by software. MCi_ADDR registers contains valid data if indicated by MCi_STATUS[ADDRV]. Table 53 defines the address register as a function of error type.

Table 53: BU error data; address register

Error Type	Address Register Bits	Description
System Data Read Error	39:6	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	39:2	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]. For all bits, 1=Enable the specified reporting mechanism.

Bits	Enable
63:2	Reserved.
1	**S_RDE_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[S_RDE_ALL] = 1.
0	**S_RDE_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[S_RDE_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]. See also MSR0000_0401 for the information about all of the MCi_STATUS registers. MSR0000_0001 alias of MSR0000_040D. See also MSRC001_0015[McStatusWrEn]. This register reports these LS errors:

Table 54: LS error signatures

Error Type	**[19:16] Error-Code-Ext**	**Error Code (see F3x48 for encoding)**						**[61] UC**	**[58] ADDRV**	**[57] PCC**	**[54:47] Synd Valid**	**[46] CECC**	**[45] UECC**	**[40] SCRUB**
		Type	**10:9 PP**	**8 T**	**7:4 RRRR**	**3:2 II/TT**	**1:0 LL**							
Read Data on Store	0000	BUS	SRC	0	DWR	MEM	LG	1	1/0	1/0	N	0	0	0
Read Data on Load	0000	BUS	SRC	0	DRD	MEM /IO	LG	1	1/0	1/0	N	0	0	0

MSR0000_040E LS Machine Check Address Register (MC3_ADDR)

Cold Reset: xxxx_xxxx_xxxx_xxxxh. See section 2.13.1 [Machine Check Architecture]. Each of the MCi_ADDR registers are written to by hardware and read-write accessible by software. MCi_ADDR registers contains valid data if indicated by MCi_STATUS[ADDRV]. The only type of error recorded by the LS machine check mechanism is a “system address out of range” or read data error for which MC3_ADDR[39:0] store the physical address. MSR0000_0000 alias of MSR0000_040E.

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)

MSR0000_0410[31:0] is a copy of [The MCA NB Control Register] F3x40; MSR0000_0410[63:32] are reserved. Only one of these registers exists in multi-core devices; see section 3.1.1 [Northbridge MSRs In Multi-Core Products].

MSR0000_0411 NB Machine Check Status Register (MC4_STATUS)

See section 2.13.1 [Machine Check Architecture]. MSR0000_0411[31:0] is a copy of [The MCA NB Status Low Register] F3x48. MSR0000_0411[63:32] is a copy of F3x4C. Only one of these registers exists in multi-core devices; see section 3.1.1 [Northbridge MSRs In Multi-Core Products].

MSR0000_0412 NB Machine Check Address Register (MC4_ADDR)

MSR0000_0412[31:0] is a copy of [The MCA NB Address Low Register] F3x50 and MSR0000_0412[63:32] is a copy of F3x54. Only one of these registers exists in multi-core devices; see section 3.1.1 [Northbridge MSRs In Multi-Core Products].

MSR0000_0413 Reserved

Reset: 0000 0000 0000 0000h. Read-only.

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 Fn0000_0001_EDX[FXSR] 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.

3.12 MSRs - MSRC001_0xxx

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

Reset: xxxx xxxx xxxx xxxxh.

PERT_CTL[3:0] are used to specify the events counted by the [The Performance Event Counter Registers (PERF_CTR[3:0])] MSRC001_00[07:04] and to control other aspects of their operation. Each performance counter supported has a corresponding event-select register that controls its operation. Section 3.14 [Performance Counter Events] shows the events and unit masks supported by the processor.

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 NB. NB events include all memory controller events (3.14.7), crossbar events (3.14.8), and link interface events (3.14.9). Monitoring of NB events should only be performed by one core. If a NB event is selected using one of the Performance Event-Select registers in any core of a multi-core processor, then a NB performance event cannot be selected in the same Performance Event Select register of any other core.

Care must be taken when measuring NB 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

63:36	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_CNT[3:0] register. The events are specified in section 3.14 [Performance Counter Events]. Some events are reserved; when a reserved event is selected, the results are undefined.

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

Reset: 0000 xxxx xxxx 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_CNT[3:0] registers can be read using a special read performance-monitoring counter instruction, RDPMC. The RDPMC instruction loads the contents of the PERF_CTR[3:0] register specified by the ECX register, into the EDX register and the EAX register.

Writing the performance counters can be useful if there is an intention for software to count a specific number of events, and then trigger an interrupt when that count is reached. An interrupt can be triggered when a performance counter overflows. Software should use the WRMSR instruction to load the count as a two's-complement negative number into the performance counter. This causes the counter to overflow after counting the appropriate number of times.

The performance counters are not assured of producing identical measurements each time they are used to measure a particular instruction sequence, and they should not be used to take measurements of very small instruction sequences. The RDPMC instruction is not serializing, and it can be executed out-of-order with respect to other instructions around it. Even when bound by serializing instructions, the system environment at the time the instruction is executed can cause events to be counted before the counter value is loaded into EDX:EAX.

Bits	Description
63:48	RAZ.
47:0	**CTR: performance counter value.** Read-write. Returns the current value of the event counter.

MSRC001_0010 System Configuration Register (SYS_CFG)

Reset: 0000 0000 0002 0601h.

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=MSRC001_001D is disabled. 1=MSRC001_001D is enabled.
20	**MtrrVarDramEn: MTRR variable DRAM enable**. Read-write. BIOS: 1. 0=[The Top Of Memory Register (TOP_MEM)] MSRC001_001A and IORRs are disabled. 1=These registers are enabled.
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. BIOS: 1. 1=Enables the RdDram and WrDram attributes in [The Fixed-Size MTRRs (MTRRfixn)] MSR0000_02[6F:68, 59, 58, 50].
17	**SysUcLockEn: system lock command enable**. Read-write. 1=Transactions within the NB support the lock command. • BIOS is recommended to set this to 1 in dual-core systems and to 0 in single core systems.

16	**ChxToDirtyDis: change to dirty disable**. Read-write. BIOS: 0. 1=Disables change-to-dirty commands, evicts line from DC instead.
15:11	Reserved.
10	**SetDirtyEnO: shared-to-dirty command for O->M state transition enable**. Read-write. Reset: 1. 1=Enables generating write probes when transitioning a cache line from Owned to Modified. This must be 1 for multi-core systems; it is recommended to be 0 for uniprocessor, single core systems.
9	**SetDirtyEnS: shared-to-dirty command for S->M state transition enable**. Read-write. Reset: 1. 1=Enables generating write probes when transitioning a cache line from Shared to Modified. This must be 1 for multi-core systems; it is recommended to be 0 for uniprocessor, single core systems.
8	Reserved.
7:5	**SysVicLimit: outstanding victim bus command limit**. Read-write. BIOS: 0h. Limits maximum number of outstanding victim bus commands.
4:0	**SysAckLimit: outstanding bus command limit**. Read-write. BIOS: 4h. Limits maximum number of outstanding bus commands. Valid values for this field are 01h through 04h. All other values may result in undefined behavior.

MSRC001_0015 Hardware Configuration Register (HWCR)

Reset: 0000 0000 0000 0000h.

Bits	Description
63:24	Reserved.
24	**TscFreqSel: TSC frequency select**. Read-write. BIOS: 1. 0= The TSC increments at the rate of the REFCLK frequency. 1=The TSC increments at the rate of the core P-state 0 COF specified by MSRC001_0064.
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 processor cores.
22:21	Reserved.
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 conditional fault for IO accesses takes priority over the IO trap mechanism described by [The IO Trap Registers (SMI_ON_IO_TRAP_[3:0])] MSRC001_00[53:50].
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]) 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. 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 "int 18" instruction. 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.

17	**Wrap32Dis: 32-bit address wrap disable**. Read-write. 1=Disable 32-bit address wrapping. Software can use Wrap32Dis to access physical memory above 4 Gbytes without switching into 64-bit mode. To do so, software should write a greater-than 4 Gbyte address to [The FS Base Register (FS_BASE)] MSRC000_0100 and [The GS Base Register (GS_BASE)] MSRC000_0101. Then it would address ±2 Gbytes from one of those bases using normal memory reference instructions with a FS or GS override prefix. However, the INVLPG, FST, and SSE store instructions generate 32-bit addresses in legacy mode, regardless of the state of Wrap32Dis.
16	Reserved.
15	**SseDis: SSE instructions disable**. Read-write. 1=Disables SSE instructions. If this is set, then CPUID Fn0000_0001_EDX[SSE, SSE2] and CPUID Fn0000_0001_ECX[SSE3] 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.
12:9	Reserved.
8	**IgnneEm: IGNNE port emulation enable**. Read-write. 1=Enable emulation of IGNNE port.
7	**DisLock: disable lock**. Read-write. 1=Disable x86 LOCK prefetch functionality.
6	**FFDis: TLB flush filter disable**. Read-write. BIOS: 1. 1=Disable TLB flush filter.
5	Reserved.
4	**INVD_WBINVD: INVD to WBINVD conversion**. Read-write. Reset: 0. 1=Convert INVD to WBINVD. BIOS is recommended to not change the state of this bit.
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 ensure proper operation.
2	Reserved.
1	**SlowFence: slow SFENCE enable**. Read-write. 1=Enable slow SFENCE.
0	**SmmLock: SMM code lock**. Read; write-1-only; cleared-by-warm-reset. 1=SMM code in the ASeg and TSeg range and the SMM registers are read-only and SMI interrupts are not intercepted in SVM.

MSRC001_00[18, 16] IO Range Registers Base (IORR_BASE[1:0])

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[39:12] & PhyMask[39:12] == PhyBase[39:12] & PhyMask[39:12].

Bits	Description
63:40	RAZ.
39:12	**PhyBase[39:12]: 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
63:40	RAZ.
39:12	**PhyMask[39:12]: 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
63:40	RAZ.
39:23	**TOM[39:23]: 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].
22:0	RAZ.

MSRC001_001D Top Of Memory 2 Register (TOM2)

Reset: X.

Bits	Description
63:40	RAZ.
39:23	**TOM2[39:23]: 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]. 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 0000h. Software is required to perform a read-modify-write in order to change any of the values in this register. Only one of these registers exists in multi-core devices; see section 3.1.1 [Northbridge MSRs In Multi-Core Products].

Bits	Description
63:59	Reserved.

58	• **EnConvertToNonIsoc**: enable conversion to non-isochronous. Read-write. BIOS: 1. 1=Convert peer-to-peer isochronous requests to non-isochronous requests (the Isoc bit in the downstream request packet is low); however, the Isoc bit in the downstream response to the requester is still set in such a case. In non-IFCM mode, the link-defined Isoc bit in the request packet is cleared as it is reflected downstream in a peer-to-peer access as well.
57:51	Reserved.
50	**DisOrderRdRsp**. Read-write. 1=Disables ordered responses to IO link read requests. See section 2.7.7 [Response Ordering].
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).
44:43	Reserved.
42	**EnaPStateSpCyc: P-state special bus cycle enable**. Read-write. 1=A P-state special bus cycle is generated for all P-state changes.
41:0	Reserved.

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[4:2]. BIOS should set these registers to the AMD-provided product name for the processor. 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[48:44] Machine Check Control Mask Registers (MC*i*_CTL_MASK)

- MSRC001_0044: Reset: 0000 0000 0000 0000h.
- MSRC001_0045: Reset: 0000 0000 0000 0000h.
- MSRC001_0046: Reset: 0000 0000 0000 0000h.
- MSRC001_0047: Reset: 0000 0000 0000 0000h.
- MSRC001_0048: Reset: 0000 0000 0000 0000h.
  - Only one of these registers exists in multi-core devices; see section 3.1.1 [Northbridge MSRs In Multi-Core Products].
  - BIOS: 0000_0000_0008_0000h. BIOS is recommended to mask HT retries (F3x40[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.

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. See section 2.13.1 [Machine Check Architecture]. For MSRC001_00[47:44]: 1=Disable error reporting for errors represented by the corresponding bit in MCi_CTL. For MSRC001_0048: 1=Disable error detection in MCi_STATUS and MCi_ADDR for errors represented by the corresponding bit in MCi_CTL.

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.

The conditional GP fault described by MSRC001_0015[IoCfgGpFault] takes priority over this trap.

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 IO access. 0=Non-Configuration IO 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.

Bits	Description
63:32	RAZ.
31:29	SBZ.
28	**C1eOnCmpHalt: C1E on dual core multi-processing halt**. Read-write. 1=When both cores of a 2 core processor have entered the halt state, the processor generates an IO cycle as specified by IORd, IOMsgData, and IOMsgAddr. See 2.4.3.1 [C1 Enhanced State (C1E)] for programming requirements. This bit is supported only for dual-core processors. Asserting this bit for single core processors may have undefined results.
27	**SmiOnCmpHalt: C1E on single core halt**. Read-write. 1=When the single core of a 1 core processor has entered the halt state, the processor generates an IO cycle as specified by IORd, IOMsgData, and IOMsgAddr. See 2.4.3.1 [C1 Enhanced State (C1E)] for programming requirements.
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]. 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

See section 2.11 [Configuration Space] for a description of MMIO configuration space. This register is accessible through F1x1[84:80] as well. Only one of these registers exists in multi-core devices; see section 3.1.1 [Northbridge MSRs In Multi-Core Products].

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>63:40</td><td>RAZ.</td></tr>
<tr><td>39:20</td><td><b>MmioCfgBaseAddr[39:20]: MMIO configuration base address bits[39:20]</b>. Read-write. Reset: X. Specifies the base address of the MMIO configuration range. The size of the MMIO configuration-space address range is defined by MSRC001_0058[BusRange] as follows. All lower order undefined bits must be 0.
<table>
<tr><th>BusRange</th><th>MmioCfgBaseAddr</th><th>BusRange</th><th>MmioCfgBaseAddr</th></tr>
<tr><td>0h</td><td>[39:20]</td><td>5h</td><td>[39:25]</td></tr>
<tr><td>1h</td><td>[39:21]</td><td>6h</td><td>[39:26]</td></tr>
<tr><td>2h</td><td>[39:22]</td><td>7h</td><td>[39:27]</td></tr>
<tr><td>3h</td><td>[39:23]</td><td>8h</td><td>[39:28]</td></tr>
<tr><td>4h</td><td>[39:24]</td><td>9-Fh</td><td>Reserved</td></tr>
</table>
</td></tr>
<tr><td>19:6</td><td>RAZ.</td></tr>
</table>

5:2	**BusRange: bus range identifier**. Read-write. Reset: X. This specifies the number of busses in the MMIO configuration space range. The size of the MMIO configuration space range varies with this field as follows: the size is 1 Mbyte times the number of busses. This field is encoded as follows: Bits Buses Bits Buses 0h 1 5h 32 1h 2 6h 64 2h 4 7h 128 3h 8 8h 256 4h 16 9-Fh Reserved
1	Reserved.
0	**Enable**. Read-write. Reset: 0. 1=MMIO configuration space is enabled.

MSRC001_0060 BIST Results Register

Reset: 0000 0000 ???? ????h. Read-only. 1=A failure was detected. See “. Built In Self Test (BIST)” on page 76.

Bits	Description
63:32	Reserved.
31	**MC**: Microcode ROM or Patch RAM failure.
30	Reserved.
29	**PDC:** Page Descriptor Cache in the bus unit.
28	**FF:** Flush Filter.
27	**L2LRU:** L2 Cache LRU array.
26	**VDB:** Victim Data Buffer in the bus unit.
25	**WDB:** Write Data Buffer in the bus unit.
24	**L2T:** L2 Cache Tag array.
23	**L2D:** L2 Cache Data array.
22	**ROBD:** Reorder Buffer Data array.
21:20	Reserved.
19	**FPRQ:** Floating Point Retire Queue.
18	**FPRR:** Floating Point Reciprocal ROM.
17	**FPCR:** Floating Point Control ROM.
16	**DCLRU:** Data Cache LRU.
15	**DCTLB2:** Data Cache L2 TLB.
14	**DCT:** Data Cache Tag Array.
13	**DCTLB1:** Data Cache L1 TLB.
12	**DCECC:** Data Cache ECC Array.
11	**DCD:** Data Cache Data Array.
10	**BSR:** Instruction Cache Branch Status Register.
9	**ICLRU:** Instruction Cache LRU.
8	**ICTLB1:** Instruction Cache L1 TLB.
7	**BH:** Instruction Cache Branch History.

6	**PDA:** Instruction Cache Predecode Array.
5	**ICD:** Instruction Cache Data Array.
4	**BSA:** Instruction Cache Branch Selector Array.
3	**BTA:** Instruction Cache Branch Target Array.
2	**ICTLB2:** Instruction Cache L2 TLB.
1	**ICST:** Instruction Cache Snoop Tag Array.
0	**ICFT**: Instruction Cache Fetch Tag Array.

MSRC001_0061 P-state Current Limit Register

See also section 2.4.2 [P-states]. 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) allowed. MSRC001_0061[PstateMaxVal] has the same value as F3xDC[PstateMaxVal]. Attempts to change MSRC001_0062[PstateCmd] to a lower-performance P-state (higher value) are clipped to the value of this field. A change to this field takes effect on the next MSRC001_0062[PstateCmd] change.
3	RAZ.
2:0	**CurPstateLimit: current P-state limit**. Read-only. Specifies the highest-performance P-state (lowest value) allowed. MSRC001_0061[CurPstateLimit] is always bounded by PstateMaxVal. Attempts to change the CurPstateLimit to the value greater (lower performance) than PstateMaxVal leaves the CurPstateLimit unchanged. CurPstateLimit = Max {000b, F3x64[HtcPstateLimit] if F3x64[HtcAct]=1}.

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[6B:64]. 0=P-state 0; 1=P-state 1; ... 7=P-state 7. P-state limits will be applied appropriately. See section 2.4.2 [P-states]. Reads from this field return the last written value, regardless of whether any limits 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. The current P-state of the core. 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].

MSRC001_00[6B:64] P-state [7:0] Registers

Reset: values vary by product.

Each of these registers specify the frequency and voltage associated with each of the core P-states.

- 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
MSRC001_0069 specifies P-state 5
MSRC001_006A specifies P-state 6
MSRC001_006B specifies P-state 7

These registers are required to be programmed to the same value in each core. See section 2.4.2 [P-states] 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[6B: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 2.4.2.6 [ACPI Processor P-state Objects]). 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
31:16	SBZ.
15:9	**CpuVid: core VID**. Read-write. See section 2.4.1 [Processor Power Planes And Voltage Control]. If this field is not programmed to the requirements specified in MSRC001_0071[MaxVid,MinVid], then undefined behavior results.
8:6	**CpuDid: core divisor ID**. Read-write. Specifies the CPU frequency divisor; see CpuFid. 0h=Divisor of 1 1h=Divisor of 2 2h=Divisor of 4 3h=Divisor of 8 4h - 7h=Reserved
5:0	**CpuFid: core frequency ID**. Read-write. Specifies the CPU frequency multiplier. • The CPU COF specified by MSRC001_00[6B:64][CpuFid,CpuDid] is ((100 MHz * (CpuFid + 08h)) / (2^CpuDid)). • The frequency specified by (100 MHz * (CpuFid + 08h)) must always be >50% of and <= 100% of the frequency specified by F3xD4[MainPllOpFreqId, MainPllOpFreqIdEn].

MSRC001_0070 COFVID Control Register

Cold reset: 0. After a warm reset this register needs to be initialized according to [The BIOS COF and VID

Requirements After Reset] 2.4.2.9.

This register includes several fields that are identical to MSRC001_00[6B:64]. It is controlled by hardware for P-state transitions. It may also be used by software to directly control the current COF or VID. Accesses to this register that result in invalid COFs or VIDs are ignored. See also section 2.4.2 [P-states].

Bits	Description
63:19	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: core VID**. Read-write. See MSRC001_00[6B:64].
8:6	**CpuDid: core divisor ID**. Read-write. See MSRC001_00[6B:64]. The PstateId field must be updated to cause a new CpuDid value to take effect.
5:0	**CpuFid: core frequency ID**. Read-write. See MSRC001_00[6B:64]. The PstateId field must be updated to cause a new CpuFid value to take effect.

MSRC001_0071 COFVID Status Register

Cold reset: values vary by product. See section 2.4.2 [P-states].

Bits	Description
63:59	Reserved.
58:56	**CurPstateLimit: current P-state limit**. Read-only. Specifies the highest-performance P-state (lowest value) allowed. Identical to MSRC001_0061[CurPstateLimit].
55	Reserved.
54:49	**MainPllOpFreqIdMax: main PLL operating frequency ID maximum**. Read-only. Specifies the maximum main PLL operating frequency supported by the processor. The maximum frequency is 100 MHz * (MainPllOpFreqIdMax + 08h), if MainPllOpFreqIdMax is greater than zero; if MainPllOpFreqIdMax = 00h, then there is no frequency limit. See F3xD4[MainPllOpFreqId].
48:42	**MinVid: minimum voltage**. Read-only. Specifies the VID code corresponding to the minimum voltage (highest VID code) that the processor drives. 00h indicates that no minimum VID code is specified. See section 2.4.1 [Processor Power Planes And Voltage Control].
41:35	**MaxVid: maximum voltage**. Read-only. Specifies the VID code corresponding to the maximum voltage (lowest VID code) that the processor drives. 00h indicates that no maximum VID code is specified. See section 2.4.1 [Processor Power Planes And Voltage Control].
34:32	**StartupPstate: startup P-state number**. Read-only. Specifies the cold reset VID and FID for the core based on the P-state number selected (see MSRC001_00[6B:64]).
31:25	**CurNbVid: current NB VID**. Read-only.
24:19	Reserved.
18:16	**CurPstate: current P-state**. Read-only. This is identical to MSRC001_0063[CurPstate].
15:9	**CurCpuVid: current core VID**. Read-only.
8:6	**CurCpuDid: current core divisor ID**. Read-only.
5:0	**CurCpuFid: current core frequency ID**. Read-only.

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]) 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)] 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[39:17] & TSegMask[39:17] == TSegBase[39:17] & TSegMask[39:17].

For example, if TSeg spans 256K bytes and starts at the 1M byte address. The MSRC001_0112[TSegBase] would be set to 0010_0000h and the MSRC001_0113[TSegMask] to FFFC_0000h (with zeros filling in for bits[16:0]). This results in a TSeg range from 0010_0000 to 0013_FFFFh.

Bits	Description
63:40	Reserved.
39:17	**TSegBase[39:17]: TSeg address range base**. IF (MSRC001_0015[SmmLock]) THEN Read-only. ELSE Read-write. ENDIF
16:0	Reserved.

MSRC001_0113 SMM TSeg Mask Register (SMMMask)

Reset: 0000 0000 0000 0000h.

See section 2.14.2 [System Management Mode (SMM)] for information about SMM.

The ASeg address range is located at a fixed address from A0000h–BFFFFh. The TSeg range is located at a variable base (specified by MSRC001_0112[TSegBase]) with a variable size (specified by MSRC001_0113[TSegMask]). These ranges provide a safe location for SMM code and data that is not readily accessible by non-SMM applications. The SMI handler can be located in one of these two ranges, or it can be located outside these ranges. These ranges must never overlap each other.

This register specifies how accesses to the ASeg and TSeg address ranges are control as follows:

- If [A, T]Valid=0, then the address range is accessed as specified by MTRRs, regardless of whether the CPU is in SMM or not.
- If [A, T]Valid=1, then:
  - If in SMM, then:
    - If [A, T]Close=0, then the accesses are directed to DRAM with memory type as specified in [A, T]MTypeDram.
    - If [A, T]Close=1, then instruction accesses are directed to DRAM with memory type as specified in [A, T]MTypeDram and data accesses are directed at MMIO space and with attributes based on [A, T]MTypeIoWc.
  - If not in SMM, then the accesses are directed at MMIO space with attributes based on [A, T]MTypeIoWc.

Bits	Description
63:40	Reserved.
39:17	**TSegMask[39:17]: TSeg address range mask**. 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
63:5	Reserved.
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 will cause 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; cleared-by-hardware. 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 and HDT 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)

Reads to this register cause a #GP. Writes 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**. Reset: X. 1=Send a RSM special cycle.
3	**smm_exit: exit SMM**. Reset: X. 1=Exit SMM.
2	**smi_cycle: send SMI special cycle**. Reset: X. 1=Send a SMI special cycle.
1	**smm_enter: enter SMM**. Reset: X. 1=Enter SMM.
0	**smm_dismiss: clear SMI**. 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_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 11h 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 11h Processors* for the definition of this field.

3.13 MSRs - MSRC001_1xxx

MSRC001_1022 Data Cache Configuration Register (DC_CFG)

All defined fields are read-write.

Bits	Description
63:16	Reserved.
15	**DIS_PF_HW_FOR_SW**. Reset: 0. 1=Disable hardware prefetches for software prefetches.
14	Reserved.
13	**DIS_HW_PF**. Reset: 0. 1=Disable hardware prefetches.
12:9	Reserved.
8	**DIS_CLR_WBTOL2_SMC_HIT**. Reset: 0. Disable clearing WBTOL2 bit in case of SMC hit.
7:0	Reserved.

MSRC001_1023 Bus Unit Configuration Register (BU_CFG)

Reset: 0000 0000 0030 2020h.

Bits	Description
63:49	Reserved.
48	**WbEnhWsbDis: disable multi-stream write combining**. Read-write. 0=The bus unit performs write combining on up to 2 indedpendent data streams. 1=The bus unit only performs write combining on a single data steam.
47:0	Reserved.

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.

Note: NB PMC events (memory controller, crossbar, link) can be lost if they occur more frequently than 1 per core clock cycle average over a 512 NB clock cycle window.

3.14.1 Floating Point Events

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; scalar operations count as one. This event is a speculative event. (See also EventSelect 0CBh). Note: Since this event includes non-numeric operations it is not suitable for measuring MFLOPs.

UnitMask	Description
01h	Add pipe ops excluding load ops.
02h	Multiply pipe ops excluding load ops.
04h	Store pipe ops excluding load ops.
08h	Add pipe load ops
10h	Multiply pipe load ops
20h	Store pipe load ops

40h	Reserved.
80h	Reserved.

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). This event is a speculative event.

3.14.2 Load/Store and TLB Events

See 3.14.4 [L2 Cache and System Interface Events] for the following:

- EventSelect 065h [Memory Requests by Type].

EventSelect 020h Segment Register Loads

The number of segment register loads performed.

UnitMask	Description
01h	ES
02h	CS
04h	SS
08h	DS
10h	FS
20h	GS
40h	HS
80h	Reserved.

EventSelect 021h Pipeline Restart Due to Self-Modifying Code

The number of pipeline restarts that were caused by self-modifying code (a store that hits any instruction that's been fetched for execution beyond the instruction doing the store).

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	Reserved.
10h	Reserved.
20h	Reserved.
40h	Reserved.
80h	Reserved.

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. Speculative. 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. A large difference between this event (with all UnitMask bits set) and EventSelect 042h would be due mainly to streaming store activity.

EventSelect 042h Data Cache Refills from L2 or System

The number of data cache refills satisfied from the L2 cache (and/or the system), 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 System
02h	Shared-state line from L2
04h	Exclusive-state line from L2
08h	Owned-state line from L2

10h	Modified-state line from L2
20h	Reserved.
40h	Reserved.
80h	Reserved.

EventSelect 043h Data Cache Refills from System

The number of L1 cache refills satisfied from the system (system memory or another core’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
20h	Reserved.
40h	Reserved.
80h	Reserved.

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). There is no provision for measuring this component by itself. 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	Description
01h	Invalid
02h	Shared
04h	Exclusive
08h	Owned
10h	Modified
20h	Reserved.
40h	Reserved.
80h	Reserved.

EventSelect 045h L1 DTLB Miss and L2 DLTB 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.

EventSelect 046h L1 DTLB and L2 DLTB Miss

The number of data cache accesses that miss in both the L1 and L2 DTLBs. This event is a speculative event.

EventSelect 047h Misaligned Accesses

The number of data cache accesses that are misaligned. These are accesses which cross an eight-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	Reserved.
08h	Reserved.
10h	Reserved.
20h	Reserved.
40h	Reserved.
80h	Reserved.

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)
08h	Reserved.
10h	Reserved.
20h	Reserved.

40h	Reserved.
80h	Reserved.

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
01h	Reserved.
02h	Data cache misses by locked instructions
04h	Reserved.
08h	Reserved.
10h	Reserved.
20h	Reserved.
40h	Reserved.
80h	Reserved.

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 were 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
04h	Reserved.
08h	Reserved.
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 System Read Responses by Coherency State

The number of responses from the system for cache refill requests. The UnitMask may be used to select specific cache coherency states. Each increment represents one 64-byte cache line transferred from the system (DRAM or another cache, including another core) 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 Quadwords Written to System

The number of quadword (8-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 eight increments; or a partial or complete Write Combining buffer flush (Sized Write), which could cause from one to eight increments.

UnitMask	Description
01h	Quadword write transfer

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	Reserved.
40h	Reserved.
80h	Reserved.

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	Reserved.
10h	Reserved.
20h	Reserved.
40h	Reserved.
80h	Reserved.

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.
04h	Reserved.
08h	Reserved.
10h	Reserved.
20h	Reserved.

40h	Reserved.
80h	Reserved.

3.14.5 Instruction Cache Events

Note: All instruction cache events are speculative events.

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 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.

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.

3.14.6 Execution Unit Events

EventSelect 026h Retired CLFLUSH Instructions

The number of CLFLUSH instructions retired.

EventSelect 027h Retired CPUID Instructions

The number of CPUID instructions retired.

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 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 were 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 that were 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 were 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. 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 were 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	Packed SSE and SSE2 instructions
08h	Scalar SSE and SSE2 instructions
10h	Reserved.
20h	Reserved.
40h	Reserved.
80h	Reserved.

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
08h	Reserved.
10h	Reserved.
20h	Reserved.
40h	Reserved.

80h	Reserved.

EventSelect 0CDh Interrupts-Masked Cycles

The number of processor cycles where interrupts are masked (EFLAGS.IF = 0). Using edge-counting with this event provides 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
10h	Reserved.
20h	Reserved.
40h	Reserved.
80h	Reserved.

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
40h	Write request
80h	Read request

EventSelect 0E1h DRAM Controller Page Table Events

The number of page table events in the local DRAM controller. This table maintains information about which DRAM pages are open. An overflow occurs when a request for a new page arrives when the page table becomes full, as the oldest entry is speculatively closed. Each occurrence reflects an access latency penalty equivalent to a page conflict.

UnitMask	Description
01h	DCT Page Table Overflow
02h	Number of stale table entry hits. (hit on a page closed too soon)

04h	Page table idle cycle limit incremented.
08h	Page table idle cycle limit decremented.
10h	Reserved.
20h	Reserved.
40h	Reserved.
80h	Reserved.

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-800, for example, would be 400 MHz).

UnitMask	Description
01h	DCT0 read-to-write turnaround
02h	DCT0 write-to-read turnaround
04h	DCT0 DIMM (chip select) turnaround
08h	DCT1 read-to-write turnaround
10h	DCT1 write-to-read turnaround
20h	DCT1 DIMM (chip select) turnaround
40h	Reserved.
80h	Reserved.

EventSelect 0E4h Memory Controller RBD Queue Events

UnitMask	Description
01h	Reserved.
02h	Reserved.
04h	F2x[1,0]94[DcqBypassMax] counter reached.
08h	Reserved.
10h	Reserved.
20h	Reserved.
40h	Reserved.
80h	Reserved.

EventSelect 0E8h Thermal Status

UnitMask	Description
01h	Number of clocks MEMHOT_L is asserted.
02h	Reserved.
04h	Number of times the HTC transitions from inactive to active.
20h	Number of clocks HTC P-state is inactive.
40h	Number of clocks HTC P-state is active.
80h	PROCHOT_L asserted by an external source and P-state change occurred.

EventSelect 0E9h CPU/IO Requests to Memory/IO

These events reflect request flow between units, as selected by the UnitMask. The UnitMask specifies the request type (CPU or IO access to IO or Memory). 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:

Request Type	CPU to Mem
CPU to Mem	A8h
CPU to IO	A4h
IO to Mem	A2h
IO to IO	A1

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.

UnitMask	Description
01h	IO to IO
02h	IO to Mem
04h	CPU to IO
08h	CPU to Mem
10h	Reserved.
20h	Reserved.
40h	Reserved.
80h	Reserved.

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)
02h	Reserved.
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)
40h	Reserved.
80h	Reserved.

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
40h	Reserved.	
80h	Reserved.	

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 a non-coherent link.

Probe results: These events reflect the results of probes sent from a memory controller to local caches. They provide an indication of the degree data and code is shared between processors (or moved between processors due to process migration). The dirty-hit events indicate the transfer of a 64-byte cache line to the requester (for a read or cache refill) or the target memory (for a write). The system bandwidth used by these, in terms of bytes per unit of time, may be calculated as 64 times the event count, divided by the elapsed time. Sized writes to memory that cover a full cache line do not incur this cache line transfer -- they simply invalidate the line and are reported as clean hits. Cache line transfers occur for Change2Dirty requests that hit cache lines in the Owned state. (Such cache lines are counted as Modified-state refills for 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 DEV Events

UnitMask	Description
01h	Reserved.
02h	Reserved.
04h	Reserved.
08h	Reserved.
10h	DEV hit
20h	DEV miss
40h	DEV error
80h	Reserved.

EventSelect 1F0h Memory Controller Requests

Sized Read/Write activity: The Sized Read/Write events reflect 32- or 64-byte transfers (as opposed to other sizes which could be anywhere between 1 and 64 bytes), from either the processor or the Hostbridge. Such accesses from the processor would be due only to write combining buffer flushes, where 32-byte accesses would reflect flushes of partially-filled buffers. Event 65h provides a count of sized write requests associated with WC buffer flushes; comparing that with counts for these events (providing there is very little Hostbridge activity at the same time) give an indication of how efficiently the write combining buffers are being used. Event 65h may also be useful in factoring out WC flushes when comparing these events with the Upstream Requests component of event ECh.

UnitMask	Description
08h	32 Bytes Sized Writes
10h	64 Bytes Sized Writes
20h	32 Bytes Sized Reads

40h	64 Byte Sized Reads

3.14.8 Crossbar Events

EventSelect 1E9h Sideband Signals and Special Cycles

UnitMask	Description
01h	HALT
02h	STOPGRANT
04h	SHUTDOWN
08h	WBINVD
10h	INVD
20h	Reserved.
40h	Reserved.
80h	Reserved.

EventSelect 1EAh Interrupt Events

UnitMask	Description
01h	Fixed and LPA
02h	LPA
04h	SMI
08h	NMI
10h	INIT
20h	STARTUP
40h	INT
80h	EOI

3.14.9 Link Events

EventSelect 0F6h Link 0 Transmit Bandwidth

UnitMask	Description
01h	Command DWORD sent
02h	Address DWORD sent
04h	Data DWORD sent
08h	Buffer release DWORD sent
10h	Nop DW sent (idle)
20h	Per packet CRC sent
40h	Reserved.

80h	Reserved.

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.