Intel® Xeon® Processor Scalable Family

Datasheet, Volume Two: Registers

July 2021

Reference Number: 614073

Notice: This document contains information on products in the design phase of development. The information here is subject to change without notice. Do not finalize a design with this information.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at Intel.com, or from the OEM or retailer.

No computer system can be absolutely secure. Intel does not assume any liability for lost or stolen data or systems or any damages resulting from such losses.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Intel® Turbo Boost Technology requires a PC with a processor with Intel Turbo Boost Technology capability. Intel Turbo Boost Technology performance varies depending on hardware, software and overall system configuration. Check with your PC manufacturer on whether your system delivers Intel Turbo Boost Technology. For more information, see http://www.intel.com/technology/turboboost

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel, the Intel logo, Intel Core, Intel SpeedStep, and Xeon are trademarks of Intel Corporation in the U.S. or its subsidiaries.

*Other names and brands may be claimed as the property of others.

Contents

Tables

Revision History

Revision Number	Description	Date
005	- Added Intel® UPI register fields	July 2021
004	- Added Intel® Core™ X-Series Processors	August 2020
003	- Added perfctrlsts_0 register information in Section 7.4	May 2019
002	- Adding Section 7 “Additional Registers” that includes: MMCFG,DMICBAR, MMCFG_BASE, MMCFG_LIMIT, TSEG, TSEG_BASE, TSEG_LIMIT, TOLM, TOHM, and VTBAR	January 2019
001	- Initial release	October 2018

§

1 Introduction

Intel® Xeon® Scalable processorThe datasheet Volume 2 provides Configuration Space Registers (CSRs).

Note: Unless specified otherwise, the “processor” will represent the following processors throughout the rest of the document:

- Intel® Xeon® Bronze 3XXX processor
- Intel® Xeon® Gold 6XXF processor
- Intel® Xeon® Platinum 6XXF processor
- Intel® Xeon® Platinum 8XXF processor
- Intel® Xeon® Silver 4XXX processor
- Intel® Xeon® Gold 5XXX processor
- Intel® Xeon® Platinum 6XXX processor
- Intel® Xeon® Platinum 8XXX processor
- Intel® Xeon® E processor family
- Intel® Xeon® W processor family
- Intel® Core i7 X-series processor family 78xx and i9-79xx series
- Intel® Core i9 X-series processor family 99xxX and i9-99xxXE series
- Intel® Core i9 X-series processor family 109xxX and i9-109xxXE series

The Intel® Xeon® processor Scalable family is the next generation of 64-bit, multi-core server processor built on a 14-nm process technology. The processor supports up to 46 bits of physical address space and 48 bits of virtual address space. The processor is designed for a platform consisting of at least one Intel® Xeon® processor Scalable family processor and the PCH. Included in this family of processors are iMC and an IIO on a single silicon die.

All processor types support up to 48 lanes of PCIe* 3.0 links capable of 8.0 GT/s, and four lanes of DMI3 and PCIe* 3.0. It features two iMCs, where each iMC supports up to three DDR4 channels with up to two DIMMs per channel.

Note: For more supported processor configurations, see the *Intel® Xeon® Processor Scalable Family/Cascade Lake Server Processor External Design Specification, Volume One: Architecture*, document number*Intel Xeon Processor Scalable Memory Family External Design Specification Volume Two: Registers, Part B*, document number 546833See also the *Intel® Xeon® Processor Scalable Family Datasheet, Volume One: Electrical*, document number 614074.

1.1 Registers Overview and Configuration Process

This is Volume 2 of the processor public document, which provides the uncore register and core MSR information for the processor. This volume documents the CSRs of each individual functional block in the uncore logic, MMIO registers for the IIO, and core MSRs. The processor contains one or more PCI devices within each functional block. The configuration registers for these devices are mapped as devices residing on the PCI bus assigned to the processor socket. CSRs are the basic hardware elements that configure the uncore logic to support various system topologies, memory configuration and densities, and hardware hooks required for RAS operations.

Note: The content contained in this volume comprehends the different processor types. Some register and field descriptions will apply only to the specific processor types. Not all features specific for each processor type have been explicitly identified in this volume, and not all features documented are available for all SKUs.

Some default values will vary based on processor type and SKU, and in most cases these are the read-only register fields that provide processor support visibility to the firmware. The firmware should not rely on the default values provided in this document, and they should instead verify these values by reading them with the firmware.

1.2 Related Publications

See the following documents for additional information.

Table 1-1. Related Publications

Document	Document Number/Location
Intel® Xeon® Processor Scalable Family Datasheet, Volume One: Electrical	614074
Intel® Xeon® Processor Scalable Family Specification Update	613537
Intel® Xeon® Processor Scalable Family Thermal Guide	613539
Intel® C620 Series Chipset Platform Controller Hub Datasheet	613516
Intel® C620 Series Chipset Thermal Mechanical Specifications and Design Guide	613522
Intel® 64 and IA-32 Architectures Software Developer's Manual Combined Volumes: 1, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D, and 4	https://software.intel.com/content/www/us/en/develop/download/intel-64-and-ia-32-architectures-sdm-combined-volumes-1-2a-2b-2c-2d-3a-3b-3c-3d-and-4.html?wapkw=325462
Intel® Virtualization Technology Specification for Directed I/O Architecture Specification	https://software.intel.com/content/www/us/en/develop/download/intel-virtualization-technology-for-directed-io-architecture-specification.html?wapkw=Intel%C2%AE%20Virtualization%20Technology%20Specification%20for%20Directed%20I%2FO%20Architecture%20Specification
Intel® Trusted Execution Technology Software Development Guide	616056

1.2.1 Terminology

Term	Description
AC	Read and Write Access Control.
ASPM	Active State Power Management.
Intel® AVX	Intel® Advanced Vector Extensions promotes legacy 128-bit SIMD instruction sets that operate on an XMM register set to use a Vector Extension (VEX) prefix and operates on 256-bit vector registers (YMM).
Intel® AVX512	The base of the 512-bit SIMD instruction extensions are referred to as Intel® Advanced Vector Extensions 512 foundation instructions. They include extensions of the Intel® AVX family of SIMD instructions but are encoded using a new encoding scheme with support for 512-bit vector registers, up to 32 vector registers in 64-bit mode, and conditional processing using opmask registers.
CA	Coherency Agent. In some cases, this is referred to as a Caching Agent, though a CA is not actually required to have a cache. It is a term used for the internal logic providing mesh interface to LLC and core. The CA is a functional unit in the CHA.
CHA	The functional module that includes the CA and home agent.
CP	Control Policy.
DMA	Direct Memory Access.
DMI3	Direct Media Interface Gen3 operating at PCIe* 3.0 speed.
DTLB	Data Translation Look-aside Buffer. Part of the processor core architecture.
DTS	Digital Thermal Sensor.
ECC	Error Correction Code.
Enhanced Intel SpeedStep® Technology	Allows the operating system to reduce power consumption when performance is not needed.
Execute Disable Bit	The execute disable bit allows the memory to be marked as executable or non-executable, when combined with a supporting operating system. If the code attempts to run in a non-executable memory, the processor raises an error to the operating system. This feature can prevent some classes of viruses or worms that exploit the buffer overrun vulnerabilities and can thus help improve the overall security of the system.
FLIT	Flow Control Unit. The Intel® UPI link layer's unit of transfer. A FLIT is made of multiple PHITS. A flit is always a fixed amount of information (192 bits).
Functional Operation	Refers to the normal operating conditions in which all processor specifications, including DC, AC, system bus, signal quality, mechanical, and thermal, are satisfied.
GSSE	Extension of the Intel® Streaming SIMD Extensions (Intel® SSE/ Intel® SSE2) floating point instruction set to 256b operands.
HA	A Home Agent orders the read and write requests to a piece of coherent memory. The HA is implemented in the CHA logic.
ICU	Instruction Cache Unit. Part of the processor core architecture.
IFU	Instruction Fetch Unit. Part of the processor core.

Term	Description
IIO	Integrated I/O Controller. An I/O controller that is integrated in the processor die. The IIO consists of the DMI3 module, PCIe* modules, and MCP (Ice Lake Server with fabric SKUs only) modules.
iMC	Integrated Memory Controller. A memory controller that is integrated in the processor die.
Intel® QuickData Technology	Intel® QuickData Technology is a platform solution designed to maximize the throughput of server data traffic across a broader range of configurations and server environments to achieve faster, scalable, and more reliable I/O.
Intel® Ultra Path Interconnect	A cache-coherent, link-based Interconnect specification for Intel processors. Also known as Intel® UPI.
Intel® 64	64-bit memory extensions to the IA-32 architecture. Further details on Intel® 64 architecture and programming models can be found at http://developer.intel.com/technology/intel64/
Intel® SPS FW	Intel® Server Platform Services Firmware. The processor uses Intel® SPS FW in server configurations.
Intel® Turbo Boost Technology	A feature that opportunistically enables the processor to run a faster frequency. This results in increased performance of both single and multi-threaded applications.
Intel® TXT	Intel® Trusted Execution Technology.
Intel® Virtualization Technology	Processor virtualization that when used in conjunction with virtual machine monitor software enables multiple, robust independent software environments inside a single platform.
Intel® VT-d	Intel® Virtualization Technology (Intel® VT) for Directed I/O. Intel® VT-d is a hardware assist, under system software (virtual machine manager or OS) control, for enabling I/O device virtualization. Intel® VT-d also brings robust security by providing protection from errant DMAs by using DMA remapping, a key feature of Intel® VT-d.
Integrated Heat Spreader (IHS)	A component of the processor package used to enhance the thermal performance of the package. Component thermal solutions interface with the processor at the IHS surface.
IOV	I/O Virtualization
IVR	Integrated Voltage Regulation.The processor supports several integrated voltage regulators.
Intel® UPI Agent	Intel® Ultra Path Interconnect (Intel® UPI) Agent. An internal logic block providing interface between internal mesh and external Intel UPI.
LLC	Last Level Cache
LRU	Least Recently Used. A term used in conjunction with cache allocation policy.
M2M	Mesh to Memory. Logic in the IMC that interfaces the IMC to the mesh.
M2PCIe*	The logic in the IIO modules that interface the modules to the mesh.
MCP	A module in the IIO enabled in Ice Lake Server with fabric that is used to interface to the on-package Intel® Omni-Path Fabric.
MESH	The on-die interconnect that connects the modules in the processor.
MESI	Modified, Exclusive, Shared, Invalid. The states used in conjunction with the cache coherency.
MLC	Mid Level Cache.

Term	Description
NCTF	Non-Critical to Function: NCTF locations are typically redundant ground or non-critical reserved, so the loss of the solder joint continuity at end-of-life conditions will not affect the overall product functionality.
NID or NodeID	Node ID or NodeID. The processor implements up to four bits of NodeID or NID.
Pcode	Pcode is microcode which is run on the dedicated microcontroller within the PCU.
PCH	Platform Controller Hub. The next generation chipset with centralized platform capabilities including the main I/O interfaces along with display connectivity, audio features, power management, manageability, security, and storage features.
PCIe* 3.0	The third generation PCIe* specification that operates at twice the speed of PCIe* 2.0 (8 Gb/s); PCIe* 3.0 is completely backward compatible with PCIe* 1.0 and 2.0.
PCIe* 2.0	PCIe* generation 2.0
PECI	Platform Environment Control Interface.
Phit	The data transfer unit on Intel® UPI at the physical layer is called a Phit or Physical Unit. A Phit will be either 20 bits or 8 bits, depending on the number of active lanes.
Processor	Includes the 64-bit cores, uncore, I/Os, and package.
Processor Core	The term “processor core” refers to Si die itself that can contain multiple execution cores. Each execution core has an instruction cache, data cache, and MLC cache. All execution cores share the L3 cache.
RAC	Read Access Control.
Rank	A unit of DRAM corresponding four to eight devices in parallel, ignoring ECC. These devices are usually, but not always, mounted on a single side of a DDR4 DIMM.
RTID	Request Transaction IDs are credits issued by the CHA to track outstanding transaction, and the RTIDs allocated to a CHA are topology dependent.
SCI	System Control Interrupt. Used in ACPI protocol.
SKU	A Stock Keeping Unit is a subset of a processor type with specific features, electrical, power, and thermal specifications. Not all features are supported on all SKUs. An SKU is based on specific use condition assumption.
Intel® SSE	Intel® Streaming SIMD Extensions.
SMBus	System Management Bus. A two-wire interface through which simple system and power management related devices can communicate with the rest of the system.
Storage Conditions	A non-operational state. The processor may be installed in a platform, in a tray, or loose. The processors may be sealed in packaging or exposed to free air. Under these conditions, processor landings should not be connected to any supply voltages, have any I/Os biased or receive any clocks. Upon exposure to “free air” (that is, unsealed packaging or a device removed from the packaging material) the processor must be handled in accordance with Moisture Sensitivity Labeling (MSL) as indicated on the packaging material.
Intel® Omni-Path	The HPC fabric enabled on the Ice lake Server with fabric. This is provided through a primary side connector on the package.
TAC	Thermal Averaging Constant.

Term	Description
TDP	Thermal Design Power.
TSOD	Temperature Sensor On DIMM.
Wolf River	The on-package component included on the Ice Lake Server with fabric that provides the Intel® Omni-Path fabric.
Uncore	The portion of the processor comprising the shared LLC cache, CHA, IMC, PCU, Utility Box (UBOC), IIO and Intel® UPI modules.
Unit Interval	Signaling convention that is binary and unidirectional. In this binary signaling, one bit is sent for every edge of the forwarded clock, whether it be a rising edge or a falling edge. If a number of edges are collected at instances t 1 , t 2 , t n ,...., t k then the UI at instance “n” is defined as: UI n = t n - t n-1.
Volume Management Device	The VMD is a new technology used to improve PCIe* management. The VMD maps the PCIe* configuration space for child devices and adapters for a particular PCIe* x16 module into its own address space, controlled by a VMD driver.
VCCIN	Primary voltage input to the voltage regulators integrated into the processor.
VSS	Processor ground.
VSSA	System agent supply for Intel® UPI and PCIe*.
VCCIO	I/O voltage supply input.
VCCD	DDR power rail.
WAC	Write Access Control.
x1, x4, x8, x16	Refers to a link or port with one, two, four or eight physical lanes.

1.4 State of Data

The data contained within this document is preliminary. It is the most accurate information available by the publication date of this document. The information in this revision of the document is based on early development data. Information may change prior to production.

§

2 Registers Overview

This is Vol 2 of the processor datasheet document that provides the CSRs of each individual functional block in the uncore logic, MMIO registers for the IIO, and core MSR information for the processor.

Note: The content in this volume comprehends multiple product types and SKUs. Some register and field descriptions will apply only to the specific product types and SKUs. Not all features specific for each processor type have been explicitly identified in this volume, and not all features documented are available for all SKUs.

Some default values will vary based on the processor type and SKU, and in most cases these are the read-only register fields that provide the processor support visibility to the firmware. The firmware should not rely on the default values provided in this document, and they should instead verify these values by reading them with the firmware.

There are two bus ranges supported for the uncore [1-0]. The bus number is configurable in the UBOX, CSR CPUBUSNO_CFG (B(30); device: 0; function: 2, offset: 0xCC). This document uses the notation: B(30) is the uncore bus 0, andB(31) is the uncore bus 1. By default, the bus number for CPUBUSNO0 is 0 and CPUBUSNO1 is 1.

2.1 Configuration Register Rules

The processor supports the following configuration register types:

- PCI CSRs: CSRs are chipset-specific registers that are located at the PCI defined address space. The processor contains PCI devices within each functional block. The configuration registers for these devices are mapped as devices residing on the PCI bus assigned to the processor socket. CSRs are the basic hardware elements that configure the uncore logic to support various system topologies, memory configuration and densities, and hardware hooks required for RAS operations.
    - When the Volume Management Device (VMD) is enabled for a particular root bus in the IIO, the VMD exposes the configuration space of its child devices through CFGBAR and the MMIO space of child devices through MEMBAR. CfgRd\Wr accesses to the child device will be dropped. A VMD driver can resurface aVMD as an additional PCI segment, allowing the child devices behind the VMD to be visible via standard methods.
- MMIO registers: These registers are mapped into the system memory map as MMIO low or MMIO high. They are accessed by any code, typically an OS driver running on the platform. This register space is introduced with the integration of some of the chipset functionality. These MMIO registers are located in the IIO module for the PCIe* segments.
- Machine Specific Registers (MSRs) are architectural and only accessed by using specific ReadMSR and WriteMSR instructions that are located in the core.

2.1.1 CSR Access

Configuration space registers are accessed via the well-known configuration transaction mechanism defined in the PCI specification, and this uses the bus:device:function number concept to address a specific device's configuration space. If initiated by a remote CPU, accesses to PCI configuration registers are achieved via NcCfgRd/Wr transactions on Intel® QuickPath Interconnect (Intel® QPI).

All configuration register accesses are accessed over the message channel through the UBOX. but itmight come from a variety of different sources:

- Local cores
- Remote cores (over Intel® QPI)

Configuration registers can be read or written in byte, WORD (16-bit), or DWORD (32-bit) quantities. Accesses larger than a DWORD to the PCIe* configuration space will result in unexpected behavior. All multi-byte numeric fields use "little-endian" ordering (that is, lower addresses contain the least significant parts of the field).

2.1.2 PCI Bus Number

In the tables shown for IIO devices (0 - 7), the PCI bus numbers are all marked as "Bus 0". This means that the actual bus number is variable depending on which socket is used. The specific bus number for all PCIe* devices in the Intel® Xeon® E5 v4 processor product family is specified in the CPUBUSNO register that exists in the I/O module's configuration space. The bus number is derived by the maximum bus range setting and processor socket number.

2.1.3 Uncore Bus Number

The PCI bus numbers are all marked as "Bus 1". This means that the actual bus number is CPUBUSNO(1), where CPUBUSNO(1) is programmable by the BIOS depending on which socket is used. The specific bus number for all PCIe* devices in the Intel® Xeon® E5 v4 processor product family is specified in the CPUBUSNO register.

2.1.4 Device Mapping

Each component in the processor is uniquely identified by a PCI bus address consisting of the bus number, device number, and function number. The device configuration is based on the PCI type 0 configuration conventions. All processor registers appear on the PCI bus assigned for the processor socket. The bus number is derived by the maximum bus range setting and processor socket number.

2.1.5 Unimplemented Devices, Functions, and Registers

- The configuration reads to unimplemented functions and devices will return all ones emulating a master abort response. There is no asynchronous error reporting that happens when a configuration read master aborts. The configuration writes to unimplemented functions and devices will return a normal response.
- The software should not attempt or rely on reads or writes to unimplemented registers or register bits. Unimplemented registers should return all zeroes when read. Writes to unimplemented registers are ignored. For configuration writes to these registers (require a completion), the completion is returned with a normal completion status (not master-aborted).

2.1.6 MSR Access

Machine specific registers are architectural and only accessed by using specific ReadMSR or WriteMSR instructions. MSRs are always accessed as a naturally aligned 4 or 8-byte quantity.

For common IA-32 architectural MSRs, see the *Intel® 64 and IA-32 Architectures Software Developer's Manual Combined Volumes: 1, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D, and 4*, available at https://software.intel.com/content/www/us/en/develop/download/intel-64-and-ia-32-architectures-sdm-combined-volumes-1-2a-2b-2c-2d-3a-3b-3c-3d-and-4.html?wapkw=325462.

2.1.7 MMIO Registers

The PCI standard provides not only configuration space registers but also registers that reside in the memory-mapped space. For PCI devices, this is typically where the majority of the driver programming occurs and the specific register definitions and characteristics are provided by the device manufacturer. Access to these registers is typically accomplished via CPU reads and writes to non-coherent (UC) or WriteCombining (WC) space. Reads and writes to memory-mapped registers can be accomplished with 1, 2, 4 or 8-byte transactions.

2.2 Register Terminology

The bits in the configuration register descriptions will have an assigned attribute from the following table. Bits without a sticky attribute are set to their default value by a hard reset.

Table 2-1. Register Attributes Definitions (Sheet 1 of 3)

Attribute	Description
RO	Read Only: These bits can only be read by software, writes have no effect. The value of the bits is determined by the hardware only.
RW	Read/Write: These bits can be read and written by the software.
RC	Read Clear variant: These bits can be read by the software, and the act of reading them automatically clears them. The HW is responsible for writing these bits, and therefore the -V modifier is implied.
W1S	Write 1 to Set: Writing a 1 to these bits will set them to 1. Writing 0 will have no effect. Reading will return indeterminate values.
WO	Write Only: These bits can only be written by microcode, reads return indeterminate values. The microcode that wants to ensure that this bit was written must read wherever the side-effect takes place.
RW-O	Read/Write Once: These bits can be read by the software. After reset, these bits can only be written by the software once, after which the bits become "Read Only".
RW-L	Read/Write Lock: These bits can be read and written by the software. The bits can be made to be "Read Only" via a separate configuration bit or other logic.
RW-KL	Read/Write Lock: These bits can be read and written by the software. The bits can be made to be "Read Only" via a separate configuration bit or other logic. Fields with this attribute also act as the locking agent for other fields.
RW1C	Read/Write 1 to Clear: These bits can be read and cleared by the software. Writing a "1" to a bit clears it, while writing a "0" to a bit has no effect.
RW0C	Read/Write 0 to Clear: These bits can be read and cleared by the software. Writing a "0" to a bit clears it, while writing a "1" has no effect.

Table 2-1. Register Attributes Definitions (Sheet 2 of 3)

Attribute	Description
ROS	RO Sticky: These bits can only be read by the software, writes have no effect. The value of the bits is determined by the hardware only. These bits are only reinitialized to their default value by a PWRGOOD reset.
RW1S	Read, Write 1 to Set: These bits can be read. Writing a "1" to a given bit will set it to "1". Writing a "0" to a given bit will have no effect. It is not possible for the software to set a bit to "0". The 1->0 transition can only be performed by the hardware. These registers are implicitly - V.
RWS	Read/Write Sticky: These bits can be read and written by the software. These bits are only reinitialized to their default value by a PWRGOOD reset.
RW1CS	Read/Write 1C Sticky: These bits can be read and cleared by the software. Writing a "1" to a bit clears it, while writing a "0" to a bit has no effect. These bits are only reinitialized to their default value by a PWRGOOD reset.
RW-LB	Read/Write Lock Bypass: Similar to RWL, these bits can be read and written by the software. The HW can make these bits "Read Only" via a separate configuration bit or other logic. However, RW-LB is a special case where the locking is controlled by the lock-bypass capability that is controlled by the lock-bypass enable bits. Each lock-bypass enable bit enables a set of configuration request sources that can bypass the lock. The requests sourced from the corresponding bypass enable bits will be lock-bypassed (RW), while requests sourced from other sources are under lock control (RO). The lock bit and bypass enable bit are generally defined with RWO attributes. Sticky can be used with this attribute (RW-SWB). These bits are only reinitialized to their default values after PWRGOOD. The lock bits may not be sticky, and it is important that they are written to after reset to guarantee that the software will not be able to change their values after a reset.
RO-FW	Read Only Forced Write: These bits are read only from the perspective of the cores.
RWS-O	If a register is both sticky and "once", then the sticky value applies to both the register value and the "once" characteristic. Only a PWRGOOD reset will reset both the value and the "once", so that the register can be written to again.
RW-V/RO-V	These bits may be modified by the hardware. The software cannot expect the values to stay unchanged. This is similar to "volatile" in software land.
RWS-V	These bits can be read or written by the software and may be modified by the hardware. The software cannot expect the values to stay unchanged. These bits are reinitialized to their default values by a PWRGOOD reset.
RWS-L	If a register is both sticky and locked, then the sticky behavior only applies to the value. The sticky behavior of the lock is determined by the register that controls the lock.
RWS-LV	These bits can be read or written by the software and may be modified by the hardware. The software cannot expect the values to stay unchanged. These bits are reinitialized to their default values by a PWRGOOD reset. If a register is both sticky and locked, then the sticky behavior only applies to the value. The sticky behavior of the lock is determined by the register that controls the lock.
SMM-RO	Read Only in SMM: These bits can only be read by software while in System Management Mode (SMM). Writes in SMM have no effect. Attempting to read or write these bits outside of SMM will cause a #GP exception to be raised.
R/SMM-W	Read/Write Only in SMM: These bits can be read by the software inside or outside of SMM but can only be written by the software while in SMM. Attempting to write these bits outside of SMM will cause a #GP exception to be raised.
SMM-RW	Read Only in SMM/Write Only in SMM: These bits can only be read and written by the software while in SMM. Attempting to write these bits outside of SMM will cause a #GP exception to be raised.

Table 2-1. Register Attributes Definitions (Sheet 3 of 3)

Attribute	Description
SMM-RW1C	Read/Write 1 to Clear in SMM: These bits can be read and cleared by the software only while in SMM. Writing a "1" to a bit clears it, while writing a "0" to a bit has no effect.
RSVD-P	Reserved - Protected: These bits are reserved for future expansion and their value must not be modified by the software. When writing these bits, the software must preserve the value read.
RSVD-Z	Reserved - Do not Care: These bits are reserved for future expansion and modifying their value has no effect. The software does not need to preserve the value read.

2.4 Notational Conventions

Hexadecimal and binary numbers

Base 16 numbers are represented by a string of hexadecimal digits followed by the character H (for example, F82EH). A hexadecimal digit is a character from the following set: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. Hexadecimal numbers can also be shown using an "x" character (for example 0x2A).

Base 2 (binary) numbers are represented by a string of ones and zeroes, sometimes followed by the character B (for example, 101B). The "B" designation is only used in situations where confusion as to the type of the number might arise.

Base 10 numbers are represented by a string of decimal digits followed by the character D (for example, 23D). The "D" designation is only used in situations where confusion as to the type of the number might arise.

§

3 iMC Configuration Registers

The iMC registers are listed here and are specific to the following:

- The Intel® Xeon® processor Scalable family implements two memory controllers each with three DDR4 memory channels and two DIMMs per channel.
    - — The iMC registersare implemented in the following bus, device, and functions:
        - Bus: B(2), device: 10,12, function: 0
        - Device 10 applies to iMC 0
        - Device 12 applies to iMC 1

For device 10 and device 12, the functions 0-5 for offsets >= 256, PCIe* extended configuration space are not designed for direct usage by the OS or device drivers, and it may not be accessible directly by the OS components such as device drivers. The PCI Capability Pointer Register (CAPPTR) is set to a value of 40h. The BIOS, the firmware, or the BMC can access these registers, combine the information obtained with system implementation specifics, and if required, make it available to the OS through the firmware or the BMC interfaces.

3.1 Device: 10 and 12, Function 0

3.1.1 pxpcap

PCIe* capability.

Type:CFGPortID:N/A
Bus: 2 Device: 10, 12 Function: 0
Offset:0x40

Bit	Attribute	Default	Description
29:25	RO	0x0	Interrupt message number (interrupt_message_number): N/A for this device
24:24	RO	0x0	Slot implemented (slot_implemented): N/A for integrated endpoints
23:20	RO	0x9	Device/port type (device_port_type): Device type is root complex integrated endpoint
19:16	RO	0x1	Capability version (capability_version): PCIe* capability is compliant with version 1.0 of the PCIe* specification. ***Note***: This capability structure is not compliant with versions beyond 1.0, since they require additional capability registers to be reserved. The only purpose for this capability structure is to make enhanced configuration space available. Minimizing the size of this structure is accomplished by reporting version 1.0 compliance and reporting that this is an integrated root port device. As such, only three DWORDs of configuration space are required for this structure.
15:8	RO	0x0	Next capability pointer (next_ptr): Pointer to the next capability. Set to 0 to indicate there are no more capability structures.
7:0	RO	0x10	Capability ID (capability_id): Provides the PCIe* capability ID assigned by the Peripheral Component Interconnect Special Interest Group* (PCI-SIG*).

3.1.2 mcmtr

Memory technology.

Type:CFGPortID:N/A
Bus:2Device:10, 12Function:0
Offset:0x87c

Bit	Attribute	Default	Description
21:18	RW_LB	0x0	CHN_DISABLE(chn_disable): Channel disable control. When set, the corresponding channel is disabled.
17:16	RW_LB	0x0	pass76(pass76): 00: Do not alter ChnAdd calculation 01: Replace ChnAdd[6] with SysAdd[6] 10: Reserved 11: Replace ChnAdd[7:6] with SysAdd[7:6]
14	RW_LB	0x0	ddr4 (ddr4): DDR4 mode
13:12	RW_LB	0x0	IMC_MODE (imc_mode): Memory mode: 00: Native DDR All others reserved
8:8	RW_LB	0x0	NORMAL (normal): 0: Training mode 1: Normal Mode
3:3	RW_LBV	0x0	DIR_EN (dir_en): If the directory disabled in SKU, this register bit is set to RO with 0 value, that is, the directory is disabled. When this bit is set to zero, iMC Error Correction Code (ECC) uses the non-directory CRC-16. If the SKU supports directory and enabled, that is, the directory is not disabled, the DIR_EN bit can be set by BIOS, iMC ECC uses CRC-15 in the first 32B code word to yield one directory bit. It is important to know that changing this bit will require BIOS to reinitialize the memory.
2:2	RW_LBV	0x0	ECC_EN (ecc_en): ECC enable. DISECC will force override this bit to 0.
1:1	RW_LBV	0x0	LS_EN (ls_en): Use lock-step channel mode if set; otherwise, independent channel mode. This field should only be set for native DDR lockstep.
0:0	RW_LB	0x0	CLOSE_PG (close_pg): Use close page address mapping if set; otherwise, open page.

3.1.3 tadwayness_[0:7]

TAD range wayness, limit, and target.

There are total of eight TAD ranges (N + P + 1 = number of TAD ranges; P = how many times channel interleave changes within the SAD ranges.).

For mirroring configuration:

For one-way interleave, channel 0-2 mirror pair: target list <0,2,x,x>, TAD ways = “00”

For one-way interleave, channel 1-3 mirror pair: target list <1,3,x,x>, TAD ways = “00”

For two-way interleave, 0-2 mirror pair and 1-3 mirror pair: target list <0,1,2,3>, TAD ways = “01”

For one-way interleave, lockstep mirroring, target list <0,2,x,x>, TAD ways = “00”

Type:CFGPortID:N/A
Bus:2Device:10, 12Function:0
Offset:0x80, 0x84, 0x88, 0x8c, 0x90, 0x94, 0x98, 0x9c, 0xa0, 0xa4, 0xa8, 0xac

Bit	Attribute	Default	Description
31:12	RW_LB	0x0	TAD_LIMIT (tad_limit): Highest address of the range in system address space, 64MB granularity, for example, TADRANGLIMIT[45:26].
11:10	RW_LB	0x0	Reserved TAD_SKT_WAY (tad_skt_way): socket interleave wayness 00 = one-way 01 = two-way 10 = four-way 11 = eight-way
9:8	RW_LB	0x0	TAD_CH_WAY (tad_ch_way): Channel interleave wayness 00 - interleave across one channel or mirror pair 01 - interleave across two channels or mirror pairs 10 - interleave across three channels 11 - interleave across four channels This parameter effectively tells the iMC how much to divide the system address by when adjusting for the channel interleave. Since both channels in a pair store every line of data, divide by “1” when interleaving across one pair and “2” when interleaving across two pairs. For Home Agent (HA), it tells how may channels to distribute the read requests across. When interleaving across one pair, this distributes the reads to two four channels, when interleaving across two pairs, this distributes the reads across four pairs. Writes always go to both channels in the pair when the read target is either channel.
7:6	RW_LB	0x0	Reserved
5:4	RW_LB	0x0	Reserved
3:2	RW_LB	0x0	Reserved
1:0	RW_LB	0x0	Reserved

3.1.4 mc_init_state_g

Initialization state for boot and training.

Type:CFGPortID:N/A
Bus:2Device:10, 12Function:0
Offset:0x8b4

Bit	Attribute	Default	Description
12:9	RWS_L	0x0	cs_oe_en:
8:8	RWS_L	0x1	MC is in SR (safe_sr): This bit indicates if it is safe to keep the MC in Self-Refresh (SR) during an MC-reset. If it is clear when the reset occurs, it means that the reset is without warning and the DDR-reset should be asserted. If set when reset occurs, it indicates that DDR is already in SR and it can keep it this way. This bit can also indicate MRC if reset without warning has occurred, and if it has, cold-reset flow should be selected. BIOS needs to clear this bit at MRC entry.
7:7	RW_L	0x0	MRC_DONE (mrc_done): This bit indicates the PCU that the MRC is done, iMC is in normal mode, ready to serve. MRC should set this bit when MRC is done, but it does not need to wait until training results are saved in BIOS flash.
5:5	RW_L	0x1	DDRIO Reset (reset_io): Training reset for DDRIO. Make sure this bit is cleared before enabling DDRIO.
3:3	RW_L	0x0	Refresh enable (refresh_enable): If cold reset, this bit should be set by BIOS after: 1) Initializing the refresh timing parameters 2) Running DDR through reset and init sequence. If warm reset or S3 exit, this bit should be set immediately after SR exit.
2:2	RW_L	0x0	DCLK enable (for all channels) (dclk_enable):
1:1	RW_L	0x1	DDR_RESET (ddr_reset): DIMM reset. Controls all channels.

3.1.5 rcomp_timer

RCOMP wait timer. Defines the time from the I/O starting to run the RCOMP evaluation until the RCOMP results are definitely ready. This counter is added to keep the determinism of the process if operated in a different mode. This register also indicates that the first RCOMP has been done and is required by the BIOS.

Type:CFGPortID:N/A
Bus:2Device:10, 12 Function:0
Offset:0x8c0

Bit	Attribute	Default	Description
31:31	RW_V	0x0	rcomp_in_progress: RCOMP in progress status bit
30:30	RW	0x0	rcomp: RCOMP start via message channel control for BIOS RCOMP start only triggered when the register bit output is changing from 0 -> 1 iMC is not responsible for clearing this bit When RCOMP is done via first_rcomp_done bit field
21:21	RW	0x0	ignore_mdll_locked_bit Ignore DDRIO MDLL lock status during RCOMP when set

Type:CFGPortID:N/A Bus:2Device:10, 12 Function:0 Offset:0x8c0			
Bit	**Attribute**	**Default**	**Description**
20:20	RW	0x0	no_mdll_fsm_override: Do not force DDRIO MDLL on during RCOMP when set
16:16	RW_LV	0x0	First RCOMP has been done in DDRIO (first_rcomp_done): This is a status bit that indicates the first RCOMP has been completed. It is cleared on reset, and set by iMC HW when the first RCOMP is completed. BIOS should wait until this bit is set before executing any DDR command
15:0	RW	0xc00	COUNT (count): DCLK cycle count that iMC needs to wait from the point it has triggered RCOMP evaluation until it can trigger the load to registers.

3.1.6 mh_ext_stat

This captures the externally asserted MEM_HOT[1:0]# and assertion detection.

Type:CFGPortID:N/A Bus:2Device:10, 12Function:0 Offset:0xe24			
Bit	**Attr**	**Default**	**Description**
1:1	RW1C	0x0	MH_EXT_STAT_1 (mh_ext_stat_1): MEM_HOT[1]# assertion status at this sense period. Set if MEM_HOT[1]# is asserted externally for this sense period. This running status bit will automatically update with the next sensed value in the next MEMHOT input sense phase.
0:0	RW1C	0x0	MH_EXT_STAT_0 (mh_ext_stat_0): MEM_HOT[0]# assertion status at this sense period. Set if MEM_HOT[0]# is asserted externally for this sense period. This running status bit will automatically update with the next sensed value in the next MEMHOT input sense phase.

3.1.7 smb_stat_[0:1]

SMBus status. This register provides the interface to the SMBus and I2C* SCL and SDA signals that is used to access the SPD EEPROM or the Thermal Sensor on DIMM (TSOD) that defines the technology, configuration, and speed of the DIMMs controlled by the iMC.

Type:CFGPortID:N/A
Bus:2Device:10, 12Function:0
Offset:0xe80, 0xe90

Bit	Attr	Default	Description
31:31	RO_V	0x0	SMB_RDO (smb_rdo): Read data valid This bit is set by the iMC when the data field of this register receives read data from the SPD/TSOD after completion of an SMBus read command. It is cleared by iMC when a subsequent SMBus read command is issued.
30:30	RO_V	0x0	SMB_WOD (smb_wod): Write operation done This bit is set by iMC when a SMBus write command has been completed on the SMBus. It is cleared by iMC when a subsequent SMBus write command is issued.
29:29	RO_V	0x0	SMB_SBE (smb_sbe): SMBus error This bit is set by iMC if an SMBus transaction (including the TSOD polling or message channel initiated SMBus access) that does not complete successfully (non-ACK has been received from slave at expected ACK slot of the transfer). If a slave device is asserting clock stretching, iMC does not have logic to detect this condition to set the SBE bit directly; however, the SMBus master will detect the error at the corresponding transaction's expected ACK slot. Once SMBUS_SBE bit is set, iMC stops issuing hardware initiated TSOD polling SMBus transactions until the SMB_SBE is cleared. iMC will not increment the SMB_STAT_x.TSOD_SA until the SMB_SBE is cleared. Manual SMBus command interface is not affected, that is, new command issue will clear the SMB_SBE like A0 silicon behavior.

Type:CFGPortID:N/A Bus:2Device:10, 12Function:0 Offset:0xe80, 0xe90			
Bit	**Attr**	**Default**	**Description**
28:28	ROS_V	0x0	SMB_BUSY (smb_busy): SMBus busy state. This bit is set by iMC while an SMBus or I2C* command (including TSOD command issued from iMC hardware) is executing. Any transaction that is completed normally or gracefully will clear this bit automatically. Setting the SMB_SOFT_RST will also clear this bit. This register bit is sticky across reset, so any surprise reset during pending SMBus operations will sustain the bit assertion across surprise warm-reset. BIOS reset handler can read this bit before issuing any SMBus transaction to determine whether a slave device may need special care to force the slave to idle state (for example, via clock override toggling SMB_CKOVRD or via induced time-out by asserting SMB_CKOVRD for 25-35 ms).
27:24	RO_V	0x7	Last issued TSOD slave address (tsod_sa): This field captures the last issued TSOD slave address. Here is the slave address and the DDR channel and DIMM slot mapping: Slave address: 0 -- Channel: Even channel; Slot #: 0 Slave address: 1 -- Channel: Even channel; Slot #: 1 Slave address: 2 -- Channel: Even channel; Slot #: 2 Slave address: 3 -- Channel: Even channel; Slot #: 3 (reserved) Slave address: 4 -- Channel: Odd channel; Slot #: 0 Slave address: 5 -- Channel: Odd channel; Slot #: 1 Slave address: 6 -- Channel: Odd channel; Slot #: 2 Slave address: 7 -- Channel: Odd channel; Slot #: 3 (reserved) This field only captures the TSOD polling slave address. During SMB error handling, software should check the hung SMB_TSOD_POLL_EN state before disabling the SMB_TSOD_POLL_EN in order to qualify whether this field is valid.
15:0	RO_V	0x0	SMB_RDATA (smb_rdata): Read data holds data read from SMBus read commands Since TSOD and EEPROM are I2C* devices and the byte order is MSByte first in a word read, reading of I2C* using word read should return SMB_RDATA[15:8] = I2C_MSB and SMB_RDATA[7:0] = I2C_LSB. If reading of I2C* using byte read, the SMB_RDATA[15:8] = do not care; SMB_RDATA[7:0] = readbyte. If there is a SMB slave connected on the bus, reading of the SMBus slave using word read returns SMB_RDATA[15:8] = SMB_LSB and SMB_RDATA[7:0] = SMB_MSB. If the software is not sure whether the target is I2C* or SMBus slave, use byte access.

3.1.8 smbcmd_[0:1]

A write to this register initiates a DIMM EEPROM access through the SMBus or I2C*.

Type:CFGPortID:N/A
Bus:2Device:10, 12Function:0
Offset:0xe84, 0xe94

Bit	Attribute	Default	Description
31:31	RW_V	0x0	SMB_CMD_TRIGGER (smb_cmd_trigger): CMD trigger: After setting this bit to "1", the SMBus master will issue the SMBus command using the other fields written in SMBCMD_[0:1] and SMBCntl_[0:1]. ***Note:*** The "-V" in the attribute implies the hardware will reset this bit when the SMBus command is being started.
30:30	RWS	0x0	SMB_PNTR_SEL (smb_pntr_sel): Pointer selection: SMBus or I2C* present pointer-based access enable when set; otherwise, use random access protocol. Hardware based TSOD polling will also use this bit to enable the pointer word read. ***Note:*** CPU hardware-based TSOD polling can be configured with pointer based access. If software manually issues SMBus transaction to other address, for example, changing the pointer in the slave device, it is the software's responsibility to restore the pointer in each TSOD before returning to hardware based TSOD polling while keeping the SMB_PNTR_SEL = 1.
29:29	RWS	0x0	SMB_WORD_ACCESS (smb_word_access): Word access: SMBus or I2C* word 2B access when set; otherwise, it is a byte access.
28:28	RWS	0x0	SMB_WRT_PNTR (smb_wrt_pntr): Bit[28:27] = 00: SMBus read Bit[28:27] = 01: SMBus write Bit[28:27] = 10: Illegal combination Bit[28:27] = 11: Write to pointer register SMBus or I2C* pointer update (byte). Bit 30 and 29 are ignored. SMBCntl_[0:1] [26] will NOT disable WrtPntr update command.
27:27	RWS	0x0	SMB_WRT_CMD (smb_wrt_cmd): When "0", it is a read command When "1", it is a write command
26:24	RWS	0x0	SMB_SA (smb_sa): Slave address: This field identifies the DIMM SPD or TSOD to be accessed.
23:16	RWS	0x0	SMB_BA (smb_ba): Bus transaction address: This field identifies the bus transaction address to be accessed. ***Note:*** In WORD access, 23:16 specifies 2B access address. In byte access, 23:16 specifies 1B access address.
15:0	RWS	0x0	SMB_WDATA (smb_wdata): Write data: Holds data to be written by SPDW commands. Since TSOD and EEPROM are I2C* devices and the byte order is MSByte first in a word write, writing of I2C* using WORD write should use SMB_WDATA[15:8] = I2C_MSB and SMB_WDATA[7:0] = I2C_LSB. If writing of I2C* using byte write, the SMB_WDATA[15:8] = do not care; SMB_WDATA[7:0] = writebyte. If an SMB slave is connected on the bus, writing of the SMBus slave using WORD write should use SMB_WDATA[15:8] = SMB_LSB and SMB_WDATA[7:0] = SMB_MSB. It is the software's responsibility to figure out the byte order of the slave access.

3.1.9 smbcntl_[0:1]

SMBus control.

Type:CFGPortID:N/A
Bus:2Device:10, 12Function:0
Offset:0xe88, 0xe98

Bit	Attribute	Default	Description
31:28	RWS	0xa	SMB_DTI (smb_dti): Device type identifier: This field specifies the device type identifier. Only devices with this device-type will respond to commands. "0011" specifies TSOD. "1010" specifies EEPROMs. "0110" specifies a write-protect operation for an EEPROM. Other identifiers can be specified to target non-EEPROM devices on the SMBus. ***Note:*** iMC-based hardware TSOD polling uses hardcoded DTI. Changing this field has no effect on the hardware based TSOD polling.
27:27	RWS_V	0x1	SMB_CKOVRD (smb_ckovrd): Clock override "0" Clock signal is driven low, overriding writing a "1" to CMD. "1" Clock signal is released high, allowing normal operation of CMD. Toggling this bit can be used to "budge" the port out of a "stuck" state. Software can write this bit to 0 and the SMB_SOFT_RST to "1"to force hung SMBus controller and the SMB slaves to idle state without using power good reset or warm reset. ***Note:*** The software needs to set the SMB_CKOVRD back to "1" after 35 ms to force slave devices to time-out in case there is any pending transaction. The corresponding SMB_STAT_x.SMB_SBE error status bit may be set if there was such pending transaction time-out (non-graceful termination). If the pending transaction was a write operation, the slave device content may be corrupted by this clock override operation. A subsequent SMB command will automatically clear the SMB_SBE. iMC added SMBus time-out control timer in B0. When the time-out control timer expires the SMBCKOVRD# will "deassert", that is, return to "1" value and clear the SMBSBE0.
26:26	RW_LB	0x1	SMB_DIS_WRT (smb_dis_wrt): Disable SMBus write Writing a "0" to this bit enables CMD to be set to "1"; writing a "1" to force CMD bit to be always 0, that is, disabling SMBus write. This bit can only be written in SMM mode. SMBus read is not affected. The I2C* write pointer update command is not affected. ***Note:*** Since the BIOS is the source to update the SMBCNTL_x register initially after reset, it is important to determine whether the SMBus can have the write capability before writing any upper bits (bit24-31) via byte-enable configuration write (or writing any bit within this register via 32b configuration write) within the SMBCNTL register.

Type:CFGPortID:N/A
Bus:2Device:10, 12Function:0
Offset:0xe88, 0xe98

Bit	Attribute	Default	Description
10:10	RW	0x0	SMB_SOFT_RST (smb_soft_rst): The SMBus software reset strobe to graceful terminate pending transaction after ACK and keep the SMB from issuing any transaction until this bit is cleared. If the slave device is hung, the software can write this bit to "1" and the SMB_CKOVRD to "0" (for more than 35 ms)to force hung the SMB slaves to time out and put it in idle state without using power good reset or warm reset. ***Note:*** The software needs to set the SMB_CKOVRD back to "1" after 35 ms to force the slave devices to time out in case there is any pending transaction. The corresponding SMB_STAT_x.SMB_SBE error status bit may be set if there was such pending transaction timeout (non-graceful termination). If the pending transaction was a write operation, the slave device content may be corrupted by this clock override operation. A subsequent SMB command will automatically clear the SMB_SBE. If the iMC HW performs an SMB timeout with the SMB_SBE_EN = 1. The software should simply clear the SMB_SBE and SMB_SOFT_RST sequentially after writing the SMB_CKOVRD = 0 and SMB_SOFT_RST = 1 asserting clock override and perform a graceful transaction termination. The hardware will automatically de-assert the SMB_CKOVRD update to "1" after the pre-configured 35 ms/65 ms timeout.
9:9	RW_LB	0x0	start_tsod_poll: This bit starts the reading of all enabled devices. The hardware will reset this bit when the SMBus polling has started.
8:8	RW_LB	0x0	SMB_TSOD_POLL_EN (smb_tsod_poll_en): TSOD polling enable "0": Disable TSOD polling and enable SPDCMD accesses. "1": Disable SPDCMD access and enable TSOD polling. It is important to make sure no pending SMBus transaction and the TSOD polling is disabled (and pending TSOD polling is drained) before changing the TSOD_POLL_EN.
7:0	RW_LB	0x0	TSOD_PRESENT for the lower and upper channels (tsod_present): The DIMM slot mask indicates whether the DIMM is equipped with the TSOD sensor. Bit 7: Must be programmed to zero. Upper channel slot #3 is not supported Bit 6: TSOD PRESENT at upper channel (ch 1 or ch 3) slot #2 Bit 5: TSOD PRESENT at upper channel (ch 1 or ch 3) slot #1 Bit 4: TSOD PRESENT at upper channel (ch 1 or ch 3) slot #0 Bit 3: Must be programmed to zero. Lower channel slot #3 is not supported Bit 2: TSOD PRESENT at lower channel (ch 0 or ch 2) slot #2 Bit 1: TSOD PRESENT at lower channel (ch 0 or ch 2) slot #1 Bit 0: TSOD PRESENT at lower channel (ch 0 or ch 2) slot #0

3.1.10 smb_tsod_poll_rate_cntr_[0:1]

Type:CFGPortID:N/A
Bus:2Device:10, 12Function:0
Offset:0xe8c, 0xe9c

Bit	Attribute	Default	Description
17:0	RW_LV	0x0	SMB_TSOD_POLL_RATE_CNTR (smb_tsod_poll_rate_cntr): TSOD poll rate counter. When it is decremented to zero, reset to zero, or written to zero, the SMB_TSOD_POLL_RATE value is loaded into this counter and the updated value appears in the next DCLK.

3.1.11 smb_period_cfg

SMBus clock period configuration.

Type:CFGPortID:N/A **Bus:2Device:10, 12Function:0** **Offset:0xea0**			
Bit	**Attribute**	**Default**	**Description**
31:16	RWS	0x445c	Reserved
15:0	RWS	0xfa0	SMB_CLK_PRD (smb_clk_prd): This field specifies the SMBus clock in number of DCLK. To generate a 50% duty cycle SCL, half of the SMB_CLK_PRD is used to generate SCL high. SCL must stay low for at least another half of the SMB_CLK_PRD before pulling high. It is recommended to program an even value in this field since the hardware is simply doing a right shift for the divided by "2" operation. The 100 KHz SMB_CLK_PRD default value is calculated based on 800 MTs (400 MHz) DCLK.

3.1.12 smb_period_cntr

SMBus clock period counter.

Type:CFGPortID:N/A **Bus:2Device:10, 12Function:0** **Offset:0xea4**			
Bit	**Attribute**	**Default**	**Description**
31:16	RO_V	0x0	SMB1_CLK_PRD_CNTR (smb1_clk_prd_cntr): SMBus #1 clock period counter for channel 23. This field is the current SMBus clock period counter value.
15:0	RO_V	0x0	SMB0_CLK_PRD_CNTR (smb0_clk_prd_cntr): SMBus #0 clock period counter for channel 01. This field is the current SMBus clock period counter value.

3.1.13 smb_tsod_poll_rate

Type:CFGPortID:N/A **Bus:2Device:10, 12Function:0** **Offset:0x1a8**			
Bit	**Attribute**	**Default**	**Description**
17:0	RWS	0x3e800	SMB_TSOD_POLL_RATE (smb_tsod_poll_rate): TSOD poll rate configuration between the consecutive TSOD accesses to the TSOD devices on the same SMBus segment. This field specifies the TSOD poll rate in number of 500 ns per CNFG_500_NANOSEC register field definition.

3.1.14 pxpcap

Type:CFGPortID:N/A **Bus:2Device:10, 12Function:1** **Offset:0x40**			
Bit	**Attribute**	**Default**	**Description**
29:25	RO	0x0	Interrupt message number (interrupt_message_number): NA for this device
24:24	RO	0x0	Slot implemented (slot_implemented): NA for integrated endpoints
23:20	RO	0x9	Device or port type (device_port_type): The device type is root complex integrated endpoint
19:16	RO	0x1	Capability version (capability_version): PCIe* capability is compliant with version 1.0 of the PCIe* Specification. ***Note:*** This capability structure is not compliant with versions beyond 1.0, since they require additional capability registers to be reserved. The only purpose for this capability structure is to make enhanced configuration space available. Minimizing the size of this structure is accomplished by reporting version 1.0 compliance and reporting that this is an integrated root port device. As such, only three DWORDS of configuration space are required for this structure.
15:8	RO	0x0	Next capability pointer (next_ptr): Pointer to the next capability. Set to 0 to indicate there are no more capability structures.
7:0	RO	0x10	Capability ID (capability_id): Provides the PCIe* capability ID assigned by PCI-SIG*.

3.1.15 spareaddresslo

Spare address low.

Always points to the lower address for the next sparing operation. This register is not affected by the HA access to the spare source rank during the HA window.

Type:CFGPortID:N/A **Bus:2Device:10, 12Function:1** **Offset:0x900**			
Bit	**Attribute**	**Default**	**Description**
31:0	RW_LV	0x0	RANKADD (rankadd): Always points to the lower address for the next sparing operation. This register will not be affected by the HA access to the spare source rank during the HA window.

3.1.16 sparectl

Type:CFGPortID:N/A Bus:2Device:10, 12Function:1 Offset:0x904			
Bit	**Attribute**	**Default**	**Description**
29:29	RW_LB	0x0	DisWPQWM (diswpqwm): Disable the WPQ level-based watermark, so that the sparing WM is only based on HaFifoWM. If DisWPQWM is clear, the spare window is started when the number of hits to the failed DIMM exceed the maximum (number of credits in WPQ not yet returned to the HA, HaFifoWM). If DisWPQWM is set, the spare window starts when the number of hits to the failed DIMM exceed HaFifoWM. In either case, if the number of hits to the failed DIMM do not hit the WM, the spare window will still start after the SPAREINTERVAL.NORMOPDUR timer expiration.
28:24	RW_LB	0x0	HaFifoWM (hafifowm): The minimum watermark for HA writes to the failed rank. Actual WM is the maximum of the WPQ credit level and HaFifoWM. When WM is hit, the HA is back-pressured, and a sparing window is started. If DisWPQWM is clear, the spare window is started when the number of hits to the failed DIMM exceed the maximum (number of credits in WPQ not yet returned to the HA, HaFifoWM). If DisWPQWM is set, the spare window starts when the number of hits to the failed DIMM exceed HaFifoWM.
23:16	RW	0x0	SCRATCH_PAD (scratch_pad): This field is available as a scratch pad.
10:8	RW_LB	0x0	DST_RANK (dst_rank): Destination logical rank used for the memory copy.
6:4	RW_LB	0x0	SRC_RANK (src_rank): Source logical rank that provides the data to be copied.
3:2	RW_LB	0x0	Channel select for the spare copy (chn_sel): Since there is only one spare-copy logic for all channels, this field selects the channel or channel-pair for the spare-copy operation. For independent channel operation: 00 = Channel 0 is selected for the spare-copy operation 01 = Channel 1 is selected for the spare-copy operation 10 = Channel 2 is selected for the spare-copy operation 11 = Channel 3 is selected for the spare-copy operation For lock-step channel operation: 0x = Channel 0 and channel 1 are selected for the spare-copy operation 1x = Channel 2 and channel 3 are selected for the spare-copy operation
0:0	RW_LBV	0x0	SPARE_ENABLE (spare_enable): Spare enable when set to “1”. The hardware clears after the sparing completion.

3.1.17 ssrstatus

Provides the status of a spare-copy memory Init operation.

Type:CFGPortID:N/A
Bus:2Device:10, 12Function:1
Offset:0x94

Bit	Attribute	Default	Description
2:2	RW1C	0x0	PATCMPLT (patcmplt): All memory has been scrubbed. The hardware sets this bit each time the patrol engine steps through all the memory locations. If the software wants to monitor 0 ---> 1 transition after the bit has been set, the software will need to clear the bit by writing "1 "to clear this bit to distinguish the next patrol scrub completion. Clearing the bit will not affect the patrol scrub operation.
1:1	RO_V	0x0	SPRCMPLT (sprcmplt): Spare operation complete. Set by the hardware once the operation is complete. The bit is cleared by the hardware when a new operation is enabled. ***Note:*** Just before the MC releases the HA block prior to the completion of the sparing operation, the iMC logic will automatically update the corresponding RIR_RNK_TGT target to reflect new DST_RANK.
0:0	RO_V	0x0	SPRINPROGRESS (sprinprogress): Spare operation in progress. This bit is set by the hardware once the operation has started. It is cleared once the operation is complete or fails.

3.1.18 scrubaddresslo

Scrub address low.

This register contains part of the address of the last patrol scrub request issued. When running the memtest, the failing address is logged in this register on memtest errors. The software can write the next address to be scrubbed into this register. The STARTSCRUB bit will then trigger the specified address to be scrubbed. Patrol scrubs must be disabled to reliably write this register.

Type:CFGPortID:N/A
Bus:2Device:10, 12Function:1
Offset:0x90C

Bit	Attribute	Default	Description
31:0	RW_LBV	0x0	RANKADD (rankadd): Contains the rank address of the last scrub issued. Can be written to specify the next scrub address with STARTSCRUB. Patrol scrubs must be disabled when writing to this field.

3.1.19 scrubaddresshi

Scrub address high.

This register pair contains part of the address of the last patrol scrub request issued. The software can write the next address into this register. Scrubbing must be disabled to reliably read and write this register. The STARTSCRUB bit will then trigger the specified address to be scrubbed.

Type:CFGPortID:N/A
Bus:2Device:10, 12Function:1
Offset:0x910

Bit	Attribute	Default	Description
17:16	RW_LBV	0x0	CHNL (chnl): Can be written to specify the next scrub address with STARTSCRUB. This register is updated with the channel address of the last scrub address issued. Patrol scrubs must be disabled when writing to this field. Only used for legacy (non-system address) patrol mode.
15:12	RW_LBV	0x0	RANK (rank): Contains the physical rank ID of the last scrub issued. Can be written to specify the next scrub address with STARTSCRUB. Restriction: Patrol scrubs must be disabled when writing to this field. Only used for legacy (non-system address) patrol mode.
11	RW_LBV	0x1	PIMARY INDICATOR (mirr_pri) Contains the primary indication when mirroring is enabled. Can be written to specify the next scrub address. Restriction: Patrol scrubs must be disabled when writing to this field. Only used for system address patrol mode.
8:0	RW_LBV	0x0	RANK ADD HI(rankaddhi): Contains the physical rank ID of the last scrub issued. Can be written to specify the next scrub address with STARTSCRUB. Patrol scrubs must be disabled when writing to this field.

3.1.20 scrubctl

This register contains the scrub control parameters and status.

Type:CFGPortID:N/A
Bus:2Device:10, 12Function:1
Offset:0x914

Bit	Attribute	Default	Description
31:31	RW_L	0x0	Scrub enable (scrub_en): Scrub enable when set.
30:30	RW_LB	0x0	Stop on complete (stop_on_cmpl): Stop patrol scrub at end of memory range. This mode is meant to be used as part of memory migration flow. Intel® Scalable Memory Interconnect (Intel® SMI) is signaled by default.
29:29	RW_LBV	0x0	patrol range complete (ptl_cmpl): When stop_on_cmpl is enabled, the patrol will stop at the end of the address range and set this bit. The patrol will resume from the beginning of the address range when this bit or stop_on_cmpl is cleared by the BIOS and the patrol scrub is still enabled by scrub_en.
28:28	RW_LB	0x0	Stop on error (stop_on_err): Stop patrol scrub on poison or uncorrectable. On poison, the patrol will log error, then stop. On uncorrectable, the patrol will convert to poison if enabled, then stop. This mode is meant to be used as part of the memory migration flow. Intel® SMI is signaled by default.
27:27	RW_LBV	0x0	patrol stopped (ptl_stopped): When stop_on_err is set, the patrol will stop on error and set this bit. The patrol will resume at the next address when this bit or stop_on_err is cleared by the BIOS and the patrol scrub is still enabled by scrub_en.
26:26	RW_LBV	0x0	SCRUBISSUED (scrubissued): When set, the scrub address registers contain the last scrub address issued.

Type:CFGPortID:N/A
Bus:2Device:10, 12Function:1
Offset:0x914

Bit	Attribute	Default	Description
25:25	RW_LB	0x0	ISSUEONCE (issueonce): When set, the patrol scrub engine will issue the address in the scrub address registers only once and stop.
24:24	RW_LBV	0x0	STARTSCRUB (startscrub): When set, the patrol scrub engine will start from the address in the scrub address registers. Once the scrub is issued, this bit is reset.
23:0	RW_LB	0x0	SCRUBINTERVAL (scrubinterval): Defines the interval in DCLKS between patrol scrub requests. The calculation for this register to get a scrub to every line in 24 hours is: ((86400)/(memory capacity/64))/cycle time of DCLK. Restrictions: It can only be changed when the patrol scrubs are disabled. Set to a minimum value of 1500.

3.1.21 spareinterval

Defines the interval between normal and sparing operations. Interval is defined in dclk.

Type:CFGPortID:N/A
Bus:2Device:10, 12Function:1
Offset:0x91c

Bit	Attribute	Default	Description
28:16	RW-LB	0x320	NUMSPARE (numspare): Sparing operation duration. The system requests will be blocked during this interval and only sparing copy operations will be serviced.
15:0	RW-LB	0xc80	Normal operation duration (normopdur): Normal operation duration. The system requests will be serviced during this interval.

3.1.22 rasenables

RAS enables register

Type:CFGPortID:N/A
Bus:2Device:10, 12Function:1
Offset:0x920

Bit	Attribute	Default	Description
0:0	RW_LB	0x0	MIRROREN (mirroren): Mirror mode enable. The channel mapping must be set up before this bit will have an effect on the iMC operation. This changes the error policy.

3.1.23 smisparectl

System management interrupt and spare control register.

Type:CFGPortID:N/A **Bus:2Device:10, 12Function:1** **Offset:0xb4**			
Bit	**Attribute**	**Default**	**Description**
17:17	RW-LB	0x0	INTRPT_SEL_PIN (intrpt_sel_pin): Enable pin signaling. When set, the interrupt is signaled via the ERROR_N[0] pin to get the attention of a BMC.
16:16	RW-LB	0x0	INTRPT_SEL_CMCI (intrpt_sel_cmci): (Corrected Machine Check Interrupt [CMCI] used as a proxy for Non-Maskable Interrupt [NMI] signaling). Set to enable NMI signaling. Clear to disable NMI signaling. If both NMI and Intel® SMI enable bits are set, then only Intel® SMI is sent.
15:15	RW-LB	0x0	INTRPT_SEL_SMI (intrpt_sel_smi): Intel® SMI enable. Set to enable Intel® SMI signaling. Clear to disable Intel® SMI signaling.

3.1.24 leaky_bucket_cfg

The leaky bucket is implemented as a 53-bit DCLK counter. The upper 42-bit of the 53-bit counter is captured in LEAKY_BUCKET_CNTR_LO and LEAKY_BUCKET_CNTR_HI registers. The carry “strobe” from the not-shown least significant 11-bit counter will trigger this 42-bit counter-pair to count. LEAKY_BUCKET_CFG contains two hot encoding thresholds LEAKY_BKT_CFG_HI and LEAKY_BKT_CFG_LO. The 42-bit counter-pair is compared with the two thresholds pair specified by LEAKY_BKT_CFG_HI and LEAKY_BKT_CFG_LO.

Type:CFGPortID:N/A
Bus:2Device:10, 12Function:1
Offset:0x928

Bit	Attribute	Default	Description
11:6	RW	0x0	LEAKY_BKT_CFG_HI (leaky_bkt_cfg_hi): This is the higher-order bit select mask of the two hot encoding thresholds. The value of this field specifies the bit position of the mask: 00h: Reserved 01h: LEAKY_BUCKET_CNTR_LO bit 1, that is, bit 12 of the full 53b counter ... 1Fh: LEAKY_BUCKET_CNTR_LO bit 31, that is, bit 42 of the full 53b counter 20h: LEAKY_BUCKET_CNTR_HI bit 0, that is, bit 43 of the full 53b counter ... 29h: LEAKY_BUCKET_CNTR_HI bit 9, that is, bit 52 of the full 53b counter 2Ah - 3F: Reserved When both counter bits selected by the LEAKY_BKT_CFG_HI and LEAKY_BKT_CFG_LO are set, the 53b leaky bucket counter will be reset and the logic will generate a primary leak strobe that is used by a 2-bit LEAKY_BKT_2ND_CNTR. LEAKY_BKT_2ND_CNTR_LIMIT specifies the value to generate the LEAK pulse, which is used to decrement the correctable error counter by "1", as shown here: LEAKY_BKT_2ND_CNTR_LIMIT LEAK pulse to decrement CE counter by "1" 00b (default): 4x primary leak strobe (four times the value programmed by the LEAKY_BKT_CFG_HI and LEAKY_BKT_CFG_LO) 01b: 1x primary leak strobe (same as the value programmed by the LEAKY_BKT_CFG_HI and LEAKY_BKT_CFG_LO) 10b: 2x primary leak strobe (two times the value programmed by the LEAKY_BKT_CFG_HI and LEAKY_BKT_CFG_LO) 11b: 3x primary leak strobe (two times the value programmed by the LEAKY_BKT_CFG_HI and LEAKY_BKT_CFG_LO) ***Note***: A value of all zeroes in the LEAKY_BUCKET_CFG register is equivalent to no leaky bucketing. The BIOS must program this register to any non-zero value before switching to normal mode.

Type:CFGPortID:N/A
Bus:2Device:10, 12Function:1
Offset:0x928

Bit	Attribute	Default	Description
5:0	RW	0x0	LEAKY_BKT_CFG_LO (leaky_bkt_cfg_lo): This is the lower-order bit select mask of the two hot encoding threshold. The value of this field specify the bit position of the mask: 00h: Reserved 01h: LEAKY_BUCKET_CNTR_LO bit 1, that is, bit 12 of the full 53b counter ... 1Fh: LEAKY_BUCKET_CNTR_LO bit 31, that is, bit 42 of the full 53b counter 20h: LEAKY_BUCKET_CNTR_HI bit 0, that is, bit 43 of the full 53b counter ... 29h: LEAKY_BUCKET_CNTR_HI bit 9, that is, bit 52 of the full 53b counter 2Ah - 3F: Reserved When both counter bits selected by the LEAKY_BKT_CFG_HI and LEAKY_BKT_CFG_LO are set, the 53b leaky bucket counter will be reset and the logic will generate a primary leak strobe that is used by a 2-bit LEAKY_BKT_2ND_CNTR. LEAKY_BKT_2ND_CNTR_LIMIT specifies the value to generate the leak pulse, which is used to decrement the correctable error counter by "1", as shown here: LEAKY_BKT_2ND_CNTR_LIMIT LEAK pulse to decrement CE counter by 1 00b (default): 4 x Primary leak strobe (four times the value programmed by the LEAKY_BKT_CFG_HI and LEAKY_BKT_CFG_LO) 01b: 1x Primary leak strobe (same as the value programmed by the LEAKY_BKT_CFG_HI and LEAKY_BKT_CFG_LO) 10b: 2x Primary leak strobe (two times the value programmed by the LEAKY_BKT_CFG_HI and LEAKY_BKT_CFG_LO) 11b: 3x Primary leak strobe (two times the value programmed by the LEAKY_BKT_CFG_HI and LEAKY_BKT_CFG_LO) ***Note***: A value of all zeroes in the LEAKY_BUCKET_CFG register is equivalent to no leaky bucketing. The MRC BIOS must program this register to any non-zero value before switching to normal mode.

3.1.25 leaky_bucket_cntr_lo

Type:CFGPortID:N/A
Bus:2Device:10, 12Function:1
Offset:0x930

Bit	Attribute	Default	Description
31:0	RW_V	0x0	Leaky bucket counter low (leaky_bkt_cntr_lo): This is the lower half of the leaky bucket counter. The full counter is actually a 53b "DCLK" counter. There is a least significant 11b of the 53b counter that is not captured in the CSR. The carry "strobe" from the not-shown least significant 11b counter will trigger this 42b counter pair to count. The 42b counter-pair is compared with the two-hot encoding threshold specified by the LEAKY_BUCKET_CFG_HI and LEAKY_BUCKET_CFG_LO pair. When the counter bits specified by the LEAKY_BUCKET_CFG_HI and LEAKY_BUCKET_CFG_LO are both set, the 53b counter is reset, and the leaky bucket logic will generate a leak strobe last for 1 DCLK.

3.1.26 leaky_bucket_cntr_hi

Type:CFGPortID:N/A Bus:2Device:10, 12Function:1 Offset:0x934			
Bit	**Attribute**	**Default**	**Description**
9:0	RW_V	0x0	Leaky bucket counter high limit (leaky_bkt_cntr_hi): This is the upper-half of the leaky bucket counter. The full counter is actually a 53b "DCLK" counter. There is a least significant 11b of the 53b counter is not captured in the CSR. The carry "strobe" from the not-shown least significant 11b counter will trigger this 42b counter pair to count. The 42b counter-pair is compared with the two-hot encoding threshold specified by the LEAKY_BUCKET_CFG_HI and LEAKY_BUCKET_CFG_LO pair. When the counter bits specified by the LEAKY_BUCKET_CFG_HI and LEAKY_BUCKET_CFG_LO are both set, the 53b counter is reset and the leaky bucket logic will generate a LEAK strobe last for 1 DCLK.

3.2 Device 10,12 Functions 2,3,4,5

3.2.1 pxpcap

Type:CFGPortID:N/A Bus:2Device:10, 12Function:2,3,4,5 Offset:0x40			
Bit	**Attribute**	**Default**	**Description**
29:25	RO	0x0	Interrupt message number (interrupt_message_number): NA for this device
24:24	RO	0x0	Slot implemented (slot_implemented): NA for integrated endpoints
23:20	RO	0x9	Device and port type (device_port_type): Device type is root complex integrated endpoint
19:16	RO	0x1	Capability version (capability_version): PCIe* capability is compliant with version 1.0 of the PCIe* specification. ***Note:*** This capability structure is not compliant with versions beyond 1.0, since they require additional capability registers to be reserved. The only purpose for this capability structure is to make enhanced configuration space available. Minimizing the size of this structure is accomplished by reporting version 1.0 compliance and reporting that this is an integrated root port device. As such, only three DWORDS of configuration space are required for this structure.
15:8	RO	0x0	Next capability pointer (next_ptr): Pointer to the next capability. Set to "0" to indicate there are no more capability structures.
7:0	RO	0x10	Capability ID (capability_id): Provides the PCIe* ccapability ID assigned by PCI-SIG*.

3.2.2 pxpenhcap

This field points to the next capability in the extended configuration space.

Type:CFGPortID:N/A **Bus:2Device:10, 12Function:2,3,4,5** **Offset:0x100**			
Bit	**Attribute**	**Default**	**Description**
31:20	RO	0x0	Next capability offset (next_capability_offset):
19:16	RO	0x0	Capability version (capability_version): Indicates there are no capability structures in the enhanced configuration space.
15:0	RO	0x0	Capability ID (capability_id): Indicates there are no capability structures in the enhanced configuration space.

3.3 Device 10,11,12 Functions 2, 6

3.3.1 pxpcap

Type:CFGPortID:N/A **Bus:2Device:10,11,12Function:2,6** **Offset:0x40**			
Bit	**Attribute**	**Default**	**Description**
7:0	RO	0x10	Capability ID (capability_id): Provides the PCIe* capability ID assigned by PCI-SIG*.

3.3.2 chn_temp_cfg

Type:CFGPortID:N/A **Bus:2Device:10,11,12Function:2,6** **Offset:0x108**			
Bit	**Attribute**	**Default**	**Description**
31:31	RW	0x1	OLTT_EN (oltt_en): Enable OLTT temperature tracking.
29:29	RW	0x0	CLTT_OR_PCODE_TEMP_MUX_SEL (cltt_or_pcode_temp_mux_sel): The TEMP_STAT byte update MUX select control to direct the source to update DIMMTEMPSTAT_[0:3][7:0]: 0: Corresponding to the DIMM TEMP_STAT byte from PCODE_TEMP_OUTPUT. 1: TSOD temperature reading from CLTT logic.
28:28	RW_O	0x1	CLTT_DEBUG_DISABLE_LOCK (cltt_debug_disable_lock): Lock bit of DIMMTEMPSTAT_[0:3][7:0]:Set this lock bit to disable the configuration write to DIMMTEMPSTAT_[0:3][7:0].
27:27	RW	0x1	Enables the thermal bandwidth throttling limit (bw_limit_thrt_en):

Type:CFGPortID:N/A Bus:2Device:10,11,12Function:2,6 Offset:0x108			
Bit	**Attribute**	**Default**	**Description**
23:16	RW	0x0	THRT_EXT (thrt_ext): Maximum number of throttled transactions to be issued during BWLIMITTF due to externally asserted MEMHOT#.
15:15	RW	0x0	THRT_ALLOW_ISOCH (thrt_allow_isoch): When this bit is zero, the MC will lower CKE during the thermal throttling, and ISOCH is blocked. When this bit is one, MC will NOT lower CKE during Thermal Throttling, and ISOCH will be allowed the base on the bandwidth throttling setting. However, setting this bit would mean more power consumption due to CKE being asserted during thermal or power throttling.
10:0	RW	0x3ff	BW_LIMIT_TF (bw_limit_tf): BW throttle window size in DCLK. ***Note:*** This value is left shifted 3 bits before being used.

3.3.3 chn_temp_stat

Type:CFGPortID:N/A Bus:2Device:10,11,12Function:2,6 Offset:0x10c			
Bit	**Attribute**	**Default**	**Description**
1:1	RW1C	0x0	Event asserted on DIMM ID 1 (ev_asrt_dimm1): Event asserted on DIMM ID 1
0:0	RW1C	0x0	Event asserted on DIMM ID 0 (ev_asrt_dimm0): Event asserted on DIMM ID 0

3.3.4 dimm_temp_oem_[0:1]

Type:CFGPortID:N/A Bus:2Device:10,11,12Function:2,6 Offset:0x110, 0x114			
Bit	**Attribute**	**Default**	**Description**
26:24	RW	0x0	TEMP_OEM_HI_HYST (temp_oem_hi_hyst): Positive going threshold hysteresis value. This value is subtracted from TEMPOEMHI to determine the point where the asserted status for that threshold will clear. Set to 00h if the sensor does not support positive-going threshold hysteresis.

Type:CFGPortID:N/A
Bus:2Device:10,11,12Function:2,6
Offset:0x110, 0x114

Bit	Attribute	Default	Description
18:16	RW	0x0	TEMP_OEM_LO_HYST (temp_oem_lo_hyst): Negative going threshold hysteresis value. This value is added to TEMPOEMLO to determine the point where the asserted status for that threshold will clear. Set to 00h if the sensor does not support negative-going threshold hysteresis.
15:8	RW	0x50	TEMP_OEM_HI (temp_oem_hi): Upper threshold value - TCase threshold at which to initiate system interrupt (Intel® SMI or MEMHOT#) at a+ going rate. ***Note:*** The default value is listed in decimal numbers. Valid range: 32 - 127 in °C. Others: Reserved.
7:0	RW	0x4b	TEMP_OEM_LO (temp_oem_lo): Lower threshold value - TCase threshold at which to initiate system interrupt (Intel® SMI or MEMHOT#) at a - going rate. ***Note:*** The default value is listed in decimal numbers. Valid range: 32 - 127 in °C. Others: Reserved.

3.3.5 dimm_temp_th_[0:2]

Type:CFGPortID:N/A
Bus:2Device:10,11,12Function:2,6
Offset:0x120, 0x124

Bit	Attribute	Default	Description
26:24	RW-LB	0x0	TEMP_THRT_HYST (temp_thrt_hyst): Positive going threshold hysteresis value. Set to 00h if the sensor does not support positive-going threshold hysteresis. This value is subtracted from TEMP_THRT_XX to determine the point where the asserted status for that threshold will clear.
23:16	RW-LB	0x5f	TEMP_HI (temp_hi): TCase threshold at which to Initiate THRTCRIT and assert THERMTRIP# valid range: 32 - 127 in °C. ***Note:*** The default value is listed in decimal numbers. FF: Disabled Others: Reserved. TEMP_HI should be programmed so it is greater than TEMP_MID.
15:8	RW	0x5a	TEMP_MID (temp_mid): TCase threshold at which to initiate THRTHI and assert valid range: 32 - 127 in °C. ***Note:*** The default value is listed in decimal numbers. FF: Disabled Others: Reserved. TEMP_MID should be programmed so it is less than TEMP_HI.
7:0	RW	0x55	TEMP_LO (temp_lo): TCase threshold at which to initiate 2x refresh or THRTMID and initiate interrupt (MEMHOT#). ***Note:*** The default value is listed in decimal.valid range: 32 - 127 in °C. FF: Disabled Others: Reserved. TEMP_LO should be programmed so it is less than TEMP_MID.

3.3.6 dimm_temp_thrt_lmt_[0:1]

All three THRT_CRIT, THRT_HI and THRT_MID are per DIMM BW limit, that is, all activities (ACT, READ, WRITE) from all ranks within a DIMM are tracked together in one DIMM activity counter. These throttle limits for high and critical are also used during scalable memory buffer thermal throttling.

Type:CFGPortID:N/A
Bus:2Device:10,11,12Function:2,6
Offset:0x130, 0x134

Bit	Attribute	Default	Description
23:16	RW-LB	0x0	THRT_CRIT (thrt_crit): Maximum number of throttled transactions (ACT, READ, WRITE) to be issued during BWLIMITTF.
15:8	RW-LB	0xf	THRT_HI (thrt_hi): Maximum number of throttled transactions (ACT, READ, WRITE) to be issued during BWLIMITTF.
7:0	RW	0xff	THRT_MID (thrt_mid): Maximum number of throttled transactions (ACT, READ, WRITE) to be issued during BWLIMITTF.

3.3.7 dimm_temp_ev_ofst_[0:1]

Type:CFGPortID:N/A
Bus:2Device:10,11,12Function:2,6
Offset:0x140, 0x144

Bit	Attribute	Default	Description
31:24	RO	0x0	TEMP_AVG_INTRVL (temp_avg_intrvl): Temperature data is averaged over this period. At the end of the averaging period (ms), the averaging process starts again. 0x1 - 0xFF Averaging data is read via TEMPDIMM STATUSREGISTER (Byte 1/2) as well as used for generating hysteresis based interrupts. 00 Instantaneous data (non-averaged) is read via TEMPDIMM STATUSREGISTER (Byte 1/2) as well as used for generating hysteresis based interrupts. ***Note:*** CPU does not support temperature averaging.
14:14	RW	0x0	Initiate THRTMID on TEMPLO (ev_thrtmid_templo): Initiate THRTMID on TEMPLO
13:13	RW	0x1	Initiate 2X refresh on TEMPLO (ev_2x_ref_templo_en): Initiate 2X refresh on TEMPLO DIMM with extended temperature range capability will need double refresh rate to avoid data loss when the DIMM temperature is above 85 °C but below 95 °C. ***Warning***: If the 2x refresh is disabled with the extended temperature range DIMM configuration, the system cooling and power thermal throttling scheme must guarantee the DIMM temperature will not exceed 85 °C.
12:12	RW	0x0	Assert MEMHOT event on TEMPHI (ev_mh_temphi_en): Assert MEMHOT# event on TEMPHI
11:11	RW	0x0	Assert MEMHOT event on TEMPMID (ev_mh_tempmid_en): Assert MEMHOT# event on TEMPMID
10:10	RW	0x0	Assert MEMHOT event on TEMPLO (ev_mh_templo_en): Assert MEMHOT# event on TEMPLO

Type:CFGPortID:N/A
Bus:2Device:10,11,12Function:2,6
Offset:0x140, 0x144

Bit	Attribute	Default	Description
9:9	RW	0x0	Assert MEMHOT event on TEMPOEMHI (ev_mh_tempoemhi_en): Assert MEMHOT# event on TEMPOEMHI
8:8	RW	0x0	Assert MEMHOT event on TEMPOEMLO (ev_mh_tempoemlo_en): Assert MEMHOT# event on TEMPOEMLO
3:0	RW	0x0	DIMM_TEMP_OFFSET (dimm_temp_offset): Temperature offset register

3.3.8 dimmtempstat_[0:1]

Type:CFGPortID:N/A
Bus:2Device:10,11,12Function:2,6
Offset:0x150, 0x154

Bit	Attribute	Default	Description
28:28	RW1C	0x0	Event asserted on TEMPHI going high (ev_asrt_temphi): Event asserted on TEMPHI going high It is assumed that each of the event assertions is going to trigger the configurable interrupt (either MEMHOT# only or both Intel® SMI and MEMHOT#) defined in bit 30 of CHN_TEMP_CFG.
27:27	RW1C	0x0	Event asserted on TEMPMID going high (ev_asrt_tempmid): Event asserted on TEMPMID going high It is assumed that each of the event assertion is going to trigger the configurable interrupt (either MEMHOT# only or both Intel® SMI and MEMHOT#) defined in bit 30 of CHN_TEMP_CFG.
26:26	RW1C	0x0	Event asserted on TEMPLO going high (ev_asrt_templo): Event asserted on TEMPLO going high It is assumed that each of the event assertions is going to trigger the configurable interrupt (either MEMHOT# only or both Intel® SMI and MEMHOT#) defined in bit 30 of CHN_TEMP_CFG.
25:25	RW1C	0x0	Event asserted on TEMPOEMLO going low (ev_asrt_tempoemlo): Event asserted on TEMPOEMLO going low It is assumed that each of the event assertions is going to trigger the configurable interrupt (either MEMHOT# only or both Intel® SMI and MEMHOT#) defined in bit 30 of CHN_TEMP_CFG.
24:24	RW1C	0x0	Event asserted on TEMPOEMHI going high (ev_asrt_tempoemhi): Event asserted on TEMPOEMHI going high It is assumed that each of the event assertions is going to trigger the configurable interrupt (either MEMHOT# only or both Intel® SMI and MEMHOT#) defined in bit 30 of CHN_TEMP_CFG.
7:0	RW_LV	0x55	DIMM_TEMP (dimm_temp): Current DIMM temperature for thermal throttling Lock by CLTT_DEBUG_DISABLE_LOCK. When the CLTT_DEBUG_DISABLE_LOCK is set, this field becomes read-only, that is, the configuration write to this byte is aborted. This byte is updated from the internal logic from a 2:1 MUX that can be selected from either CLTT temperature or from the corresponding temperature registers output (PCODE_TEMP_OUTPUT) updated from the pcode. The MUX selection is controlled by CLTT_OR_PCODE_TEMP_MUX_SEL defined in the CHN_TEMP_CFG register. Valid range from 0 to 127 that is, 0 °C to +127 °C. Any negative value read from TSOD is forced to "0". The TSOD decimal point value is also truncated to the integer value.

3.3.9 thrt_pwr_dimm_[0:1]

bit[10:0]: Maximum number of transactions (ACT, READ, WRITE) to be allowed during the 1 microsecond throttling timeframe per power throttling.

Type:CFGPortID:N/A
Bus:2Device:10,11,12Function:2,6
Offset:0x190, 0x192

Bit	Attribute	Default	Description
15:15	RW	0x1	THRT_PWR_EN (thrt_pwr_en): bit[15]: set to "1" to enable the power throttling for the DIMM.
11:0	RW	0xfff	Power throttling control (thrt_pwr): bit[11:0]: Maximum number of transactions (ACT, READ, WRITE) to be allowed (per DIMM) during the 1 microsecond throttling timeframe per power throttling.

3.4 Device 10,12 Functions 3,7

3.4.1 correrrcnt_0

Per-rank corrected error counters.

Type:CFGPortID:N/A
us:2Device:10,12Function:3,7
Offset:0x104

Bit	Attribute	Default	Description
31:31	RW1CS	0x0	RANK 1 OVERFLOW (overflow_1): The corrected error count for this rank has been overflowed. Once set, it can only be cleared via a write from the BIOS.
30:16	RWS_LV	0x0	RANK 1 CORRECTABLE ERROR COUNT (cor_err_cnt_1): The corrected error count for this rank. The hardware automatically clears this field when the corresponding OVERFLOW_x bit is changing from 0 to 1.
15:15	RW1CS	0x0	RANK 0 OVERFLOW (overflow_0): The corrected error count for this rank has been overflowed. Once set, it can only be cleared via a write from the BIOS.
14:0	RWS_LV	0x0	RANK 0 CORRECTABLE ERROR COUNT (cor_err_cnt_0): The corrected error count for this rank. The hardware automatically clears this field when the corresponding OVERFLOW_x bit is changing from 0 to 1.

3.4.2 correrrcnt_1

Per-rank corrected error counters.

Type:CFGPortID:N/A
us:2Device:10,12Function:3,7
Offset:0x108

Bit	Attribute	Default	Description
31:31	RW1CS	0x0	RANK 3 OVERFLOW (overflow_3): The corrected error count has crested over the limit for this rank. Once set, it can only be cleared via a write from the BIOS.
30:16	RWS_LV	0x0	RANK 3 COR_ERR_CNT (cor_err_cnt_3): The corrected error count for this rank.
15:15	RW1CS	0x0	RANK 2 OVERFLOW (overflow_2): The corrected error count has crested over the limit for this rank. Once set, it can only be cleared via a write from the BIOS.
14:0	RWS_LV	0x0	RANK 2 COR_ERR_CNT (cor_err_cnt_2): The corrected error count for this rank.

3.4.3 correrrcnt_2

Per-rank corrected error counters.

Type:CFGPortID:N/A
Bus:1Device:20,21,23Function:2,3
Offset:0x10c

Bit	Attribute	Default	Description
31:31	RW1CS	0x0	RANK 5 OVERFLOW (overflow_5): The corrected error count has crested over the limit for this rank. Once set, it can only be cleared via a write from the BIOS.
30:16	RWS_LV	0x0	RANK 5 COR_ERR_CNT (cor_err_cnt_5): The corrected error count for this rank.
15:15	RW1CS	0x0	RANK 4 OVERFLOW (overflow_4): The corrected error count has crested over the limit for this rank. Once set, it can only be cleared via a write from the BIOS.
14:0	RWS_LV	0x0	RANK 4 COR_ERR_CNT (cor_err_cnt_4): The corrected error count for this rank.

3.4.4 correrrcnt_3

Per-rank corrected error counters.

Type:CFGPortID:N/A
Bus:1Device:20,21,23Function:2,3
Offset:0x110

Bit	Attribute	Default	Description
31:31	RW1CS	0x0	RANK 7 OVERFLOW (overflow_7): The corrected error count for this rank.

<table>
<tr><td colspan="4"><b>Type:CFGPortID:N/A
Bus:1Device:20,21,23Function:2,3
Offset:0x110</b></td></tr>
<tr><th>Bit</th><th>Attribute</th><th>Default</th><th>Description</th></tr>
<tr><td>30:16</td><td>RWS_LV</td><td>0x0</td><td>RANK 7 COR_ERR_CNT_7 (cor_err_cnt_7):
The corrected error count for this rank.</td></tr>
<tr><td>15:15</td><td>RW1CS</td><td>0x0</td><td>RANK 6 OVERFLOW (overflow_6):
The corrected error count has crested over the limit for this rank. Once set, it can only be cleared via a write from the BIOS.</td></tr>
<tr><td>14:0</td><td>RWS_LV</td><td>0x0</td><td>RANK 6 COR_ERR_CNT (cor_err_cnt_6):
The corrected error count for this rank.</td></tr>
</table>

3.4.5 correrrthrshld_0

This register holds the per-rank corrected error thresholding value.

<table>
<tr><td colspan="4"><b>Type:CFGPortID:N/A
us:2Device:10,12Function:3,7
Offset:0x11c</b></td></tr>
<tr><th>Bit</th><th>Attribute</th><th>Default</th><th>Description</th></tr>
<tr><td>30:16</td><td>RW-LB</td><td>0x7fff</td><td>RANK 1 COR_ERR_TH (cor_err_th_1):
The corrected error threshold for this rank that will be compared to the per-rank corrected error counter.</td></tr>
<tr><td>14:0</td><td>RW-LB</td><td>0x7fff</td><td>RANK 0 COR_ERR_TH (cor_err_th_0):
The corrected error threshold for this rank that will be compared to the per-rank corrected error counter.</td></tr>
</table>

3.4.6 correrrthrshld_1

This register holds the per-rank corrected error thresholding value.

<table>
<tr><td colspan="4"><b>Type:CFGPortID:N/A
us:2Device:10,12Function:3,7
Offset:0x120</b></td></tr>
<tr><th>Bit</th><th>Attribute</th><th>Default</th><th>Description</th></tr>
<tr><td>30:16</td><td>RW-LB</td><td>0x7fff</td><td>RANK 3 COR_ERR_TH (cor_err_th_3):
The corrected error threshold for this rank that will be compared to the per-rank corrected error counter.</td></tr>
<tr><td>14:0</td><td>RW-LB</td><td>0x7fff</td><td>RANK 2 COR_ERR_TH (cor_err_th_2):
The corrected error threshold for this rank that will be compared to the per-rank corrected error counter.</td></tr>
</table>

3.4.7 correrrthrshld_2

This register holds the per-rank corrected error thresholding value.

Type:CFGPortID:N/A **us:2Device:10,12Function:3,7** **Offset:0x124**			
Bit	**Attribute**	**Default**	**Description**
30:16	RW-LB	0x7fff	RANK 5 COR_ERR_TH (cor_err_th_5): The corrected error threshold for this rank that will be compared to the per-rank corrected error counter.
14:0	RW-LB	0x7fff	RANK 4 COR_ERR_TH (cor_err_th_4): The corrected error threshold for this rank that will be compared to the per-rank corrected error counter.

3.4.8 correrrthrshld_3

This register holds the per-rank corrected error thresholding value.

Type:CFGPortID:N/A **us:2Device:10,12Function:3,7** **Offset:0x128**			
Bit	**Attribute**	**Default**	**Description**
30:16	RW-LB	0x7fff	RANK 7 COR_ERR_TH (cor_err_th_7): The corrected error threshold for this rank that will be compared to the per-rank corrected error counter.
14:0	RW-LB	0x7fff	RANK 6 COR_ERR_TH (cor_err_th_6): The corrected error threshold for this rank that will be compared to the per-rank corrected error counter.

3.4.9 correrrorstatus

Per-rank corrected error status. These bits are reset by the BIOS.

Type:CFGPortID:N/A **us:2Device:10,12Function:3,7** **Offset:0x134**			
Bit	**Attribute**	**Default**	**Description**
31:24	RW_V	0x0	ddr4crc_rank_log: This field gets set with 1'b1 if the corresponding rank detected DDR4 CRC in one of its write data. This will be cleared by the BIOS.
7:0	RW1C	0x0	ERR_OVERFLOW_STAT (err_overflow_stat): This 8-bit field is the per-rank error over-threshold status bits. The organization is as follows: Bit 0: Rank 0 Bit 1: Rank 1 Bit 2: Rank 2 Bit 3: Rank 3 Bit 4: Rank 4 Bit 5: Rank 5 Bit 6: Rank 6 Bit 7: Rank 7 ***Note:*** The register tracks which rank has reached or exceeded the corresponding CORRERRTHRSHLD threshold settings.

3.4.10 leaky_bkt_2nd_cntr_reg

Type:CFGPortID:N/A **us:2Device:10,12Function:3,7** **Offset:0x138**			
Bit	**Attribute**	**Default**	**Description**
31:16	RW	0x0	LEAKY_BKT_2ND_CNTR_LIMIT(leaky_bkt_2nd_cntr_limit): Secondary leaky bucket counter limit (2b per DIMM). This register defines the secondary leaky bucket counter limit for all eight logical ranks within the channel. The counter logic will generate the secondary LEAK pulse to decrement the rank's correctable error counter by "1" when the corresponding rank leaky bucket rank counter rolls over at the predefined counter limit. The counter increments at the primary leak pulse from the LEAKY_BUCKET_CNTR_LO and LEAKY_BUCKET_CNTR_HI logic. Bit[31:30]: Rank 7 secondary leaky bucket counter limit Bit[29:28]: Rank 6 secondary leaky bucket counter limit Bit[27:26]: Rank 5 secondary leaky bucket counter limit Bit[25:24]: Rank 4 secondary leaky bucket counter limit Bit[23:22]: Rank 3 secondary leaky bucket counter limit Bit[21:20]: Rank 2 secondary leaky bucket counter limit Bit[19:18]: Rank 1 secondary leaky bucket counter limit Bit[17:16]: Rank 0 secondary leaky bucket counter limit The value of the limit is defined as the following: 0: The leak pulse is generated one DCLK after the primary LEAK pulse is asserted. 1: The LEAK pulse is generated one DCLK after the counter rolls over at 1. 2: The LEAK pulse is generated one DCLK after the counter rolls over at 2. 3: The LEAK pulse is generated one DCLK after the counter rolls over at 3.
15:0	RW_V	0x0	LEAKY_BKT_2ND_CNTR (leaky_bkt_2nd_cntr): Per-rank secondary leaky bucket counter (2b per rank) bit [15:14]: Rank 7 secondary leaky bucket counter bit [13:12]: Rank 6 secondary leaky bucket counter bit [11:10]: Rank 5 secondary leaky bucket counter bit [9:8]: Rank 4 secondary leaky bucket counter bit [7:6]: Rank 3 secondary leaky bucket counter bit [5:4]: Rank 2 secondary leaky bucket counter bit [3:2]: Rank 1 secondary leaky bucket counter bit [1:0]: Rank 0 secondary leaky bucket counter

3.4.11 devtag_cntl_[0:7]

SDDC usage model.

When the number of correctable errors (CORRERRCNT_x) from a particular rank exceeds the corresponding threshold (CORRERRTHRSHLD_y), the hardware will generate a LINK interrupt and log (and preserve) the failing device in the FailDevice field. The SMM software will read the failing device on the particular rank. The software then sets the EN bit to enable substitution of the failing device or rank with the parity from the rest of the devices in line.

For independent channel configuration, each rank can tag once. Up to eight ranks can be tagged.

For lock-step channel configuration, only one x8 device can be tagged per rank pair. The SMM software must identify which channel should be tagged for this rank and only set the valid bit for the channel from the channel-pair.

There is no hardware logic to report an incorrect programming error. An unpredicable error or silent data corruption will be the consequence of such a programming error.

If the rank-sparing is enabled, it is recommend to prioritize the rank-sparing before triggering the device tagging due to the nature of the device tagging, which would drop the correction capability, and any subsequent ECC error from this rank would cause an uncorrectable error.

Type:CFGPortID:N/A
us:2Device:10,12Function:3,7
Offset:0x140, 0x141, 0x142, 0x143, 0x144, 0x145, 0x146, 0x147

Bit	Attribute	Default	Description
7:7	RWS_L	0x0	Device tagging enable for this rank (en): Device tagging SDDC enable for this rank. Once set, the parity device of the rank is used for the replacement device content. After tagging, the rank will no longer have the “correction” capability. The ECC error “detection” capability will not degrade after setting this bit. For the lock-step channel configuration, only one x8 device can be tagged per rank pair. The SMM software must identify which channel should be tagged for this rank and only set the corresponding DEVTAG_CNTL_x.EN bit for the channel that contains the failed device. The DEVTAG_CNTL_x.EN on the other channel of the corresponding rank must not be set.
5:0	RWS_V	0x3f	Fail device ID for this rank (faildevice): Once set, the parity device of the rank is used for the replacement device content. After tagging, the rank will no longer have the "correction" capability. The ECC error "detection" capability will not degrade after setting this bit. ***Warning:*** For the lockstep channel configuration, only one x8 device can be tagged per rank pair. The SMM software must identify which channel should be tagged for this rank and only set the corresponding DEVTAG_CNTL_x.EN bit for the channel contains the failed device. The DEVTAG_CNTL_x.EN on the other channel of the corresponding rank must not be set. DDDC: (EX processor only) On DDDC supported systems, the BIOS has the option to enable SDDC in conjunction with DDDC_CNTL:SPARING to enable faster sparing with SDDC substitution. This field is cleared by the HW on completion of DDDC sparing.

§

4 Intel® UPI Registers

The Intel® UPI module is the coherent communication interface between processors. The number of supported Intel® UPI links varies per processor type. Bus: B(3), device: 16-14, function: 0 (Intel® UPI)

4.1 Bus: 3, Device: 16, 14, Function: 3

4.1.1 ktimiscstat

Intel® UPI miscellaneous status.

Bus: B(3)	Device: 16-14	Function: 3	Offset: D4
Bit	**Attribute**	**Default**	**Description**
31:3	RSVD-Z	0000000 0 h	Reserved — Do not care.
2:0	RO-V	3h	**kti_rate** — This reflects the supported current Intel® UPI rate setting into the PLL 100 - 9.6 GT/s. 101 - 10.4GT/s Other - Reserved ***Note:*** The default value of 3'b011 does not reflect the actual Intel® UPI rate. Reads of this register field will always report one of the legal defined values previously mentioned.

4.1.2 ktilp0

Intel® UPI link parameter 0.

Bus: B(3)	Device: 16-14	Function: 0	Offset: 94
Bit	**Attribute**	**Default**	**Description**
19:16	RO-V	0h	**base_nodeid** — Parameter bits from the peer agent. Cleared on any LL initialization. The NodeID of the sending socket.
12:8	RO-V	00h	**sending_port** — Parameter bits from the peer agent. Cleared on any LL initialization. The processor die port number of the sending port. Legal values are 0-2.

4.1.3 ktipcsts

Intel® UPI Pcode status. This register is used by PCode to store the link training status.

Bus: B(3)	Device: 16-14	Function: 0	Offset: 120
Bit	**Attribute**	**Default**	**Description**
4	RW	0h	**ll_status_valid** — Bit indicates the valid training status logged from PCode to the BIOS.

§

5 Configuration Agent (UBOX) Registers

The UBOX handles transactions such as register accesses, interrupt flows, lock flows, and events. This includes transactions like the register accesses, interrupt flows, lock flows, and events.The UBOX houses the coordination for the performance architecture, scratchpad, and semaphore registers.

5.1 Bus: 0, Device: 8, Function: 0

5.1.1 Vendor ID (VID)

PCI vendor ID register.

Bus: B(0)	Device: 8	Function: 0	Offset: 0
Bit	**Attribute**	**Default**	**Description**
15:0	RO	8086h	**Vendor_Identification_Number** — The value is assigned by PCI-SIG* to Intel.

5.1.2 Device Identification (DID)

PCI device identification number

Bus: B(0)	Device: 8	Function: 0	Offset: 2
Bit	**Attribute**	**Default**	**Description**
15:0	RO	2014h	**Device_Identification_Number** —

5.1.3 CPUNODEID

Node ID configuration register

Bus: B(0)	Device: 8	Function: 0	Offset: C0
Bit	**Attribute**	**Default**	**Description**
31:16	RSVD-Z	0000h	Reserved — Do not care.
15:13	RW-LB	0h	**NodeCtrlId** — Node ID of the node controller. Set by the BIOS.
12:10	RW-LB	0h	**LgcNodeId** — NodeID of the legacy socket.
9:8	RSVD-Z	0h	Reserved — Do not care.
7:5	RW-LB	0h	**LockNodeId** — NodeID of the lock master.
4:3	RSVD-Z	0h	Reserved — Do not care.
2:0	RW-LB	0h	**LclNodeId** — Node ID of the local socket.

5.1.4 IntControl

Interrupt configuration register.

Bus: B(0)	Device: 8	Function: 0	Offset: C8
Bit	**Attribute**	**Default**	**Description**
31:19	RSVD-Z	0000h	Reserved — Do not care.
18	RW-LB	0h	**LogFlatClustOvrEn** — 0: IA32 The logical flat or cluster mode bit is locked as a read-only bit. 1: IA32 LThe logical flat or cluster mode bit may be written by the SW, and the values written by the xTPR update are ignored. For one time override of the IA32 logical flat or cluster mode value, return this bit to its default state after the bit is changed. Leaving this bit as "1" will prevent an automatic update of the filter.
17	RW-LBV	0h	**LogFltClustMod** — Set by the BIOS to indicate if the OS is running the logical flat or logical cluster mode. This bit can also be updated by IntPrioUpd messages. This bit reflects the setup of the filter at any given time. 0 - flat, 1 - cluster.
16	RW-LB	0h	**ClastChkSmpMod** — 0: Disable checking for Logical_APICID[31:0] being non-zero when sampling a flat or cluster mode bit in the IntPrioUpd message as part of setting bit 1 in this register. 1: Enable the previously mentioned checking.
15:11	RSVD-Z	00h	Reserved — Do not care.
10:8	RW	0h	**HashModCtr** — Indicates the hash mode control for the interrupt control. Select the hush function for the vector-based hash mode interrupt redirection control: 000 selects bits 7:4/5:4 for the vector cluster or flat algorithm 001 selects bits 6:3/4:3 010 selects bits 4:1/2:1 011 selects bits 3:0/1:0 Other - Reserved
7	RSVD-Z	0h	Reserved — Do not care.
6:4	RW	0h	**RdrModSel** — Selects the redirection mode used for MSI interrupts with the lowest-priority delivery mode. The following schemes are used: 000: Fixed priority - Selects the first enabled Advanced Programmable Interrupt Controller (APIC) in the cluster. 001: Redirect last - Last vector selected (applicable only in extended mode). 010: Hash vector - Selects the first enabled APIC in a round robin manner starting form the hash of the vector number. Default: Fixed priority.
3:2	RSVD-Z	0h	Reserved — Do not care.
1	RW-LB	0h	**ForceX2APIC** — Write: 1: Forces the system to move into X2APIC mode. 0: No affect.
0	RW-LB	1h	**xApicEn** — Set this bit if you would like the extended XAPIC configuration to be used. This bit can be written directly, and it can also be updated using xTPR messages

5.1.5 GIDNIDMAP

Mapping between the group ID and the NodeID.

Bus: B(0)	Device: 8	Function: 0	Offset: D4
Bit	**Attribute**	**Default**	**Description**
31:24	RSVD-Z	00h	Reserved — Do not care.
23:21	RW-LB	0h	**NodeId7** — NodeID for group id 7
20:18	RW-LB	0h	**NodeId6** — NodeID for group 6
17:15	RW-LB	0h	**NodeId5** — NodeID for group 5
14:12	RW-LB	0h	**NodeId4** — NodeID for group id 4
11:9	RW-LB	0h	**NodeId3** — NodeID for group 3
8:6	RW-LB	0h	**NodeID2** — NodeID for group Id 2
5:3	RW-LB	0h	**NodeId1** — NodeID for group Id 1
2:0	RW-LB	0h	**NodeId0** — NodeID for group 0

5.1.6 UBOXErrSts

This is the error status register in the UBOX, and it covers most of the interrupt related errors.

Bus: B(0)	Device: 8	Function: 0	Offset: C8
Bit	**Attribute**	**Default**	**Description**
31:24	RSVD-Z	00h	Reserved — Do not care.
23:18	RWS-V	00h	**Msg_Ch_Tkr_TimeOut** — Message channel tracker timeout. This error occurs when any NP request does not receive a response in 4K cycles.
17	RWS-V	0h	**Msg_Ch_Tkr_Err** — Message channel tracker error. This error occurs in such cases where there is an illegal broadcast port ID access to the message channel.
16	RW-V	0h	**SMI_delivery_valid** — SMI interrupt delivery status valid, write 1'b0 to clear valid status.
15:8	RO-V	00h	**reserved** — Reserved
7	RWS-V	0h	**MasterLockTimeOut** — Master lock timeout received by the UBOX.
6	RWS-V	0h	**SMITimeOut** — SMI timeout received by the UBOX.
5	RWS-V	0h	**CFGWrAddrMisAligned** — MMCFG write address misalignment received by the UBOX. All MMCFG access must be less than or equal to 4B in length and cannot cross a 4B boundary. When the UBOX sees a misaligned MMCFG access, it will abort the transaction.
4	RWS-V	0h	**CFGRdAddrMisAligned** — MMCFG read address misalignment received by the UBOX. All MMCFG access must be less than or equal to 4B in length and cannot cross a 4B boundary. When the UBOX sees a misaligned MMCFG access, it will abort the transaction.
3	RWS-V	0h	**UnsupportedOpcode** — Unsupported opcode received by the UBOX.
2	RWS-V	0h	**PoisonRsvd** — The UBOX received a poisoned transaction.
1	RWS-V	0h	**SMISrciMC** — SMI is caused due to an indication from the iMC.
0	RWS-V	0h	**SMISrcUMC** — This is a bit that indicates that an SMI was caused due to a locally generated UMC.

5.2 Bus: 0, Device: 8, Function: 2 VID

PCI vendor ID register.

Bus: B(0)	Device: 8	Function: 2	Offset: 0
Bit	**Attribute**	**Default**	**Description**
15:0	RO	8086h	**Vendor_Identification_Number** — The value is assigned by PCI-SIG* to Intel.

5.2.1 DID

PCI device identification number.

Bus: B(0)	Device: 8	Function: 2	Offset: 2
Bit	**Attribute**	**Default**	**Description**
15:0	RO	2016h	**Device_Identification_Number** —

5.2.2 CPUBUSNO

Bus number configuration.

Bus: B(0)	Device: 8	Function: 2	Offset: CC
Bit	**Attribute**	**Default**	**Description**
31:24	RW-LB	03h	**CPUBUSNO3** — Bus number 3
23:16	RW-LB	02h	**CPUBUSNO2** — Bus number 2
15:8	RW-LB	01h	**CPUBUSNO1** — Bus number 1
7:0	RW-LB	00h	**CPUBUSNO0** — Bus number 0

5.2.3 CPUBUSNO1

Bus number configuration 1.

Bus: B(0)	Device: 8	Function: 2	Offset: D0
Bit	**Attribute**	**Default**	**Description**
31:16	RSVD-Z	0000h	Reserved — Do not care.
15:8	RW-LB	05h	**CPUBUSNO5** — Bus number 5
7:0	RW-LB	04h	**CPUBUSNO4** — Bus number 4

5.2.4 SMICtrl

SMI generation control

Bus: B(0)	Device: 8	Function: 2	Offset: D8
Bit	**Attribute**	**Default**	**Description**
31:29	RSVD-Z	0h	Reserved — Do not care.
28	RW-LB	0h	**SMIDis4** — Disable generation of SMI from CSMI from MsgCh
27	RW-LB	0h	**SMIDis3** — Disable generation of SMI from message channel
26	RW-LB	1h	**SMIDis2** — Disable generation of SMI for lock timeout, cfg write mis-align access, and cfg read mis-align access
25	RW-LB	0h	**SMIDis** — Disable generation of SMI
24	RSVD-P	0h	Reserved — Protected
23:20	RSVD-Z	0h	Reserved — Do not care
19:0	RSVD-P	00000h	Reserved — Protected

§

6 PCU Registers

The PCU is a dedicated controller that provides power and thermal management for the processor. The PCU implements a PECI interface for out-of-band management. The PCU consists of a dedicated microcontroller, ROM and RAM for Pcode (PCU microcode), HW state machines, I/O registers for interfacing to the microcontroller, and interfaces to the hardware units in the processor.

6.1 Bus: B1, Device: 30, Function: 0

6.1.1 VID

PCI vendor ID register.

Bus: B(1)	Device: 30	Function: 0	Offset: 0
Bit	**Attribute**	**Default**	**Description**
15:0	RO	8086h	Vendor_Identification_Number — The value is assigned by PCI-SIG* to Intel.

6.1.2 DID

PCI device identification number.

Bus: B(1)	Device: 30	Function: 0	Offset: 2
Bit	**Attribute**	**Default**	**Description**
15:0	RO	2080h	Device_Identification_Number —

6.1.3 PACKAGE_ENERGY_STATUS

The package energy consumed by the entire CPU (including core and uncore). The counter will wrap around and continue counting when it reaches its limit.

Bus: B(1)	Device: 30	Function: 0	Offset: 90
Bit	**Attribute**	**Default**	**Description**
31:0	RO-V	00000000h	**DATA** — Refer to MSR 611h which this is a mirror of for the description.

6.1.4 MEM_TRML_TEMPERATURE_REPORT_0

This register is used to report the thermal status of the memory. The channel's maximum temperature field is used to report the maximal temperature of all ranks.

MEM_TRML_TEMPERATURE_REPORT_0 is used for channel temperature of DIMMs under IMC0.

Bus: B(1) Device: 30Function: 0Offset: 94			
Bit	**Attribute**	**Default**	**Description**
31:24	RSVD-P	00h	Reserved — Protected
23:16	RO-V	00h	**Channel2_Max_Temperature** — Temperature in °C
15:8	RO-V	00h	**Channel1_Max_Temperature** — Temperature in °C
7:0	RO-V	00h	**Channel0_Max_Temperature** — Temperature in °C

6.1.5 MEM_TRML_TEMPERATURE_REPORT_1

This register is used to report the thermal status of the memory. The channel's maximum temperature field is used to report the maximal temperature of all ranks.

MEM_TRML_TEMPERATURE_REPORT_1 is used for the channel temperature of DIMMs under iMC1.

Bus: B(1) Device: 30Function: 0Offset: 98			
Bit	**Attribute**	**Default**	**Description**
31:24	RSVD-P	00h	Reserved — protected.
23:16	RO-V	00h	**Channel2_Max_Temperature** — Temperature in °C
15:8	RO-V	00h	**Channel1_Max_Temperature** — Temperature in °C
7:0	RO-V	00h	**Channel0_Max_Temperature** — Temperature in °C

6.1.6 MEM_TRML_TEMPERATURE_REPORT_2

This register is used to report the thermal status of the memory. The channel's maximum temperature field is used to report the maximal temperature of all ranks.

Bus: B(1) Device: 30Function: 0Offset: 9C			
Bit	**Attribute**	**Default**	**Description**
31:24	RSVD-P	00h	Reserved — protected.
23:16	RO-V	00h	**Channel2_Max_Temperature** — Temperature in °C
15:8	RO-V	00h	**Channel1_Max_Temperature** — Temperature in °C
7:0	RO-V	00h	**Channel0_Max_Temperature** — Temperature in °C

6.1.7 PACKAGE_TEMPERATURE

Package temperature in ^{o}C. This field is updated by the FW.

Bus: B(1)Device: 30Function: 0Offset: C8			
Bit	**Attribute**	**Default**	**Description**
31:8	RSVD-Z	000000h	Reserved — Do not care
7:0	RO-V	00h	**DATA** — Package temperature in ^{o}C

6.1.8 TEMPERATURE_TARGET

Legacy register holding temperature related constants for platform use.

Bus: B(1)Device: 30Function: 0Offset: E4			
Bit	**Attribute**	**Default**	**Description**
31:28	RSVD-Z	0h	Reserved — Do not care
27:24	RW	0h	**TJ_MAX_TCC_OFFSET** — Refer to MSR 1A2h which this is a mirror of for the description
23:16	RO-V	00h	**REF_TEMP** — Refer to MSR 1A2h which this is a mirror of for the description
15:8	RO-V	00h	**FAN_TEMP_TARGET_OFST** — Refer to MSR 1A2h which this is a mirror of for the description
7:0	RSVD-Z	00h	Reserved — Do not care

6.2 Bus: B(1), Device: 30, Function: 2

6.2.1 VID

PCI vendor ID register.

Bus: B(1)	Device: 30	Function: 2	Offset: 0
Bit	**Attribute**	**Default**	**Description**
15:0	RO	8086h	**Vendor_Identification_Number** — The value is assigned by PCI-SIG* to Intel.

6.2.2 DID

PCI device identification number.

Bus: B(1)	Device: 30	Function: 2	Offset: 2
Bit	**Attribute**	**Default**	**Description**
15:0	RO	2082h	**Device_Identification_Number** —

6.2.3 PCICMD

PCI Command Register

Bus: B(1)Device: 30Function: 2Offset: 4			
Bit	**Attr**	**Default**	**Description**
15:11	RSVD-Z	00h	Reserved — don't care.
10	RO	0h	**INTx_Disable** — N/A for these devices
9	RO	0h	**Fast_Back_to_Back_Enable** — Not applicable to PCI Express and is hardwired to 0
8	RO	0h	**SERR_Enable** — This bit has no impact on error reporting from these devices
7	RO	0h	**IDSEL_Stepping_Wait_Cycle_Control** — Not applicable to internal devices. Hardwired to 0.
6	RO	0h	**Parity_Error_Response** — This bit has no impact on error reporting from these devices
5	RO	0h	**VGA_palette_snoop_Enable** — Not applicable to internal devices. Hardwired to 0.
4	RO	0h	**Memory_Write_and_Invalidate_Enable** — Not applicable to internal devices. Hardwired to 0.
3	RO	0h	**Special_Cycle_Enable** — Not applicable. Hardwired to 0.
2	RO	0h	**Bus_Master_Enable** — Hardwired to 0 since these devices don't generate any transactions
1	RO	0h	**Memory_Space_Enable** — Hardwired to 0 since these devices don't decode any memory BARs
0	RO	0h	**IO_Space_Enable** — Hardwired to 0 since these devices don't decode any IO BARs

6.2.4 PCISTS

PCI Status Register

Bus: B(1)		Device: 30	Function: 2 Offset: 6
Bit	**Attr**	**Default**	**Description**
15	RO	0h	**Detected_Parity_Error** — This bit is set when the device receives a packet on the primary side with an uncorrectable data error (including a packet with poison bit set) or an uncorrectable address/control parity error. The setting of this bit is regardless of the Parity Error Response bit (PERRE) in the PCICMD register.
14	RO	0h	**Signaled_System_Error** — Hardwired to 0
13	RO	0h	**Received_Master_Abort** — Hardwired to 0
12	RO	0h	**Received_Target_Abort** — Hardwired to 0
11	RO	0h	**Signaled_Target_Abort** — Hardwired to 0
10:9	RO	0h	**DEVSEL_Timing** — Not applicable to PCI Express. Hardwired to 0.
8	RO	0h	**Master_Data_Parity_Error** — Hardwired to 0
7	RO	0h	**Fast_Back_to_Back** — Not applicable to PCI Express. Hardwired to 0.
6	RO	0h	Reserved — Reserved
5	RO	0h	**x66MHz_capable** — Not applicable to PCI Express. Hardwired to 0.

4	RO	0h	**Capabilities_List** — This bit indicates the presence of a capabilities list structure. When set to 1, indicates the register at 34h provides an offset into the function.
3	RO	0h	**INTx_Status** — Reflects the state of the INTA# signal at the input of the enable/ disable circuit. This bit is set by HW to 1 when the INTA# is asserted. This bit is reset by HW to 0 after the interrupt is cleared.
2:0	RSVD-Z	0h	Reserved — don't care.

6.2.5 RID

PCIe header Revision ID Register

Bus: B(1)		**Device: 30**	**Function: 2** **Offset: 8**
Bit	**Attr**	**Default**	**Description**
7:0	ROS-V	00h	**revision_id** — Reflects the Uncore Revision ID after reset. Reflects the Compatibility Revision ID after BIOS writes 0x69 to any RID register in the processor uncore.

6.2.6 CCR

PCIe header ClassCode register

Bus: B(1)Device: 30Function: 2Offset: 9			
Bit	**Attr**	**Default**	**Description**
23:16	RO-V	08h	**base_class** — The value changes dependent upon the dev-func accessed.
15:8	RO-V	80h	**sub_class** — The value changes dependent upon the dev-func accessed.
7:0	RO-V	00h	**register_level_programming_interface** —

6.2.7 CLSR

PCI Cache Line Size Register

Bus: B(1)		**Device: 30**	**Function: 2** **Offset: C**
Bit	**Attr**	**Default**	**Description**
7:0	RO	00h	**Cacheline_Size** — Size of Cacheline

6.2.8 PLAT

PCI Latency Timer

Bus: B(1)		**Device: 30**	**Function: 2** **Offset: D**
Bit	**Attr**	**Default**	**Description**
7:0	RO	00h	**Primary_Latency_Timer** — Not applicable to PCI-Express. Hardwired to 00h.

6.2.9 HDR

PCI Header Type

Bus: B(1)Device: 30Function: 2Offset: E			
Bit	**Attr**	**Default**	**Description**
7	RO	1h	**Multi_function_Device** — This bit defaults to 1b since all these devices are multi-function
6:0	RO	00h	**configuration_layout** — Type 0 header

6.2.10 BIST

PCI BIST Register

Bus: B(1)Device: 30Function: 2Offset: F			
Bit	**Attr**	**Default**	**Description**
7:0	RO	00h	**BIST_Tests** — Not supported. Hardwired to 00h

6.2.11 SVID

PCI Subsystem Vendor ID Register

Bus: B(1)		Device: 30	Function: 2 Offset: 2C
Bit	**Attr**	**Default**	**Description**
15:0	RWS-O	8086h	**Subsystem_Vendor_Identification_Number** — The default value specifies Intel but can be set to any value once after reset.

6.2.12 SDID

PCI Subsystem device ID Register

Bus: B(1)		Device: 30	Function: 2 Offset: 2E
Bit	**Attr**	**Default**	**Description**
15:0	RWS-O	0000h	**Subsystem_Device_Identification_Number** — Assigned by the subsystem vendor to uniquely identify the subsystem

6.2.13 CAPPTR

PCI Capability Pointer Register

Bus: B(1)		Device: 30	Function: 2 Offset: 34
Bit	**Attr**	**Default**	**Description**
7:0	RO	00h	**Capability_Pointer** — Points to the first capability structure for the device which is the PCIe capability.

6.2.14 INTL

PCI Interrupt Line Register

Bus: B(1)		Device: 30	Function: 2 Offset: 3C
Bit	**Attr**	**Default**	**Description**
7:0	RO	00h	**Interrupt_Line** — N/A for these devices

6.2.15 INTPIN

PCI Interrupt Pin Register

Bus: B(1)		Device: 30	Function: 2 Offset: 3D
Bit	**Attr**	**Default**	**Description**
7:0	RO	00h	Interrupt_Pin — N/A since these devices do not generate any interrupt on their own

6.2.16 MINGNT

PCI Min Grant Register

Bus: B(1)		Device: 30	Function: 2 Offset: 3E
Bit	**Attr**	**Default**	**Description**
7:0	RO	00h	**MGV** — The device does not burst as a PCI compliant master.

6.2.17 MAXLAT

PCI Max Latency Register

Bus: B(1)		Device: 30	Function: 2 Offset: 3F
Bit	**Attr**	**Default**	**Description**
7:0	RO	00h	**MLV** — The device has no specific requirements for how often it needs to access the PCI bus.

6.2.18 DRAM_ENERGY_STATUS

The DRAM energy consumed by all the DIMMS in all the channels. The counter will wrap around and continue counting when it reaches its limit.

ENERGY_UNIT for DRAM domain is 15.3uJ.

The data is updated by Pcode and is read only for all the SW.

6.2.19 PACKAGE_RAPL_PERF_STATUS

Bus: B(1)	Device: 30	Function: 2	Offset: 7C
Bit	**Attribute**	**Default**	**Description**
31:0	RO-V	00000000h	**DATA** — Refer to MSR 619h which this is a mirror of for the description.

This register is used to report the package power limit violations.

6.2.20 DRAM_POWER_INFO

Bus: B(1)Device: 30Function: 2Offset: 88			
Bit	**Attribute**	**Default**	**Description**
31:0	RO-V	00000000h	**PWR_LIMIT_THROTTLE_CTR** — Refer to MSR 613h which this is a mirror of for the description.

Bus: B(1)Device: 30Function: 2Offset: A8			
Bit	**Attribute**	**Default**	**Description**
63	RW-KL	0h	**Lock** — Refer to MSR 61Ch which this is a mirror of for the description.
62:55	RSVD-Z	00h	Reserved — Do not care.
54:48	RW-L	28h	**DRAM_MAX_WIN** — Refer to MSR 61Ch which this is a mirror of for the description.
47	RSVD-Z	0h	Reserved — Do not care.
46:32	RW-L	0258h	**DRAM_MAX_PWR** — Refer to MSR 61Ch which this is a mirror of for the description.
31	RSVD-Z	0h	Reserved — Do not care.
30:16	RW-L	0078h	**DRAM_MIN_PWR** — Refer to MSR 61Ch which this is a mirror of for the description.
15	RSVD-Z	0h	Reserved — Do not care.
14:0	RW-L	0118h	**DRAM_TDP** — Refer to MSR 61Ch which this is a mirror of for the description.

6.2.21 DRAM_RAPL_PERF_STATUS

This register is used by Pcode to report the DRAM plane power limit violations in the platform.

Bus: B(1)Device: 30Function: 2Offset: D8			
Bit	**Attribute**	**Default**	**Description**
31:0	RO-V	00000000h	**PWR_LIMIT_THROTTLE_CTR** — Refer to MSR 61Bh which this is a mirror of for the description.

6.2.22 THERMTRIP_CONFIG

This register is used to configure whether the thermtrip signal only carries the processor trip information, or if it carries the mem trip information as well. The register will be used by the HW to enable the ORing of the memtrip information into the thermtrip OR tree.

Bus: B(1)Device: 30Function: 2Offset: F8			
Bit	**Attribute**	**Default**	**Description**
31:4	RSVD-Z	0000000h	Reserved — Do not care.
3:1	RSVD-P	0h	Reserved — Protected.
0	RW-LB	0h	**EN_MEMTRIP** — If set to “1”, PCU will OR in the memtrip information into the thermtrip OR tree. If set to “0”, the PCU will ignore the memtrip information and thermtrip will just have the processor indication. Expect the BIOS to enable this in Phase 4.

§

7 Additional Registers

7.1 MMCFG_Base

MMCFG address base.

Bus: RootBus0 Device: 5Function: 0Offset: 90h			
Bit	**Attribute**	**Default**	**Description**
63:51	RV	0	Reserved — Do not care.
50:26	RW-LB	1FFFFFFh	MMCFG_BASE MMCFG base address Indicates the base address that is aligned to a 64-MB boundary.
25:0	RV	0	Reserved — Do not care.

7.2 MMCFG_LIMIT

MMCFG address limit.

Bus: RootBus0 Device: 5Function: 0Offset: 98h			
Bit	**Attribute**	**Default**	**Description**
63:51	RV	0	Reserved — Do not care.
50:26	RW-LB	0000000h	MMCFG_LIMIT MMCFG limit address Indicates the limit address that is aligned to a 64-MB boundary. Any inbound request to the following region targets the MMCFG region and is aborted. MMCFG.BASE <= Addr[31:26] <= MMCFG.LIMIT Address bits [25:0] are ignored and may be any value. Address bits [63:32] must be 0. Setting the MMCFG.BASE greater than MMCFG.LIMIT disables this region.
25:0	RV	0	Reserved — Do not care.

7.3 TSEG

TSeg address range.

Bus: RootBus0 Device: 5Function: 0Offset: A8h			
Bit	**Attribute**	**Default**	**Description**
63:52	RW-LB	000h	LIMIT TSeg limit address Indicates the limit address that is aligned to a 1-MB boundary. Any access that falls within TSEG.BASE[31:20] <= Addr[31:20] <= TSEG.LIMIT[31:20] is considered to target the Tseg region and IIO aborts it. The address bits 19:0 are ignored and not compared. The result is that BASE[19:0] is effectively 00000h and LIMIT is effectively FFFFFh. Setting the TSEG.BASE greater than the limit disables this region. Setting BASE[31:20] = LIMIT[31:0] opens a 1-MB window due to address bits [19:0] being ignored for this comparison.
51:32	RV	0h	Reserved.
31:20	RW-LB	FE0h	BASE TSeg base address Indicates the base address that is aligned to a 1-MB boundary. Bits [31:20] correspond to A[31:20] address bits.
19:0	RV	0	Reserved.

7.4 Top of Low Memory (TOLM)

Top of low memory address.

Bus: RootBus0 Device: 5Function: 0Offset: D0h			
Bit	**Attribute**	**Default**	**Description**
63:26	RW-LB	00h	ADDR TOLM address Indicates the top of the low DRAM memory that is aligned to a 64-MB boundary. A 32-bit transaction that satisfies “0 <= Address[31:26] <= TOLM[31:26]” is a transaction towards the main memory.
25:0	RV	0h	Reserved.

7.5 Top of High Memory (TOHM)

Top of high memory address.

Bus: RootBus0 Device: 5Function: 0Offset: D8h			
Bit	**Attribute**	**Default**	**Description**
63:26	RW-LB	00h	ADDR TOHM address Indicates the limit of an aligned 64-MB granular region that decodes >4 GB addresses towards the system DRAM memory. A 64-bit transaction that satisfies "4G <= A[63:26] <= TOHM[63:26]" is a transaction towards the main memory. This register is programmed once at boot time and does not change after that, including during quiesce flows.
25:0	RV	0h	Reserved.

7.6 VTBAR

Base address register for Intel® VT-d registers.

Bus: RootBus0 Device: 5Function: 0Offset: 180h			
Bit	**Attribute**	**Default**	**Description**
31:13	RW-LB	00000h	vtd_chipset_base_address Intel® VT-d base address Provides an aligned 8K base address for the IIO registers relating to Intel® VT-d. All inbound accesses to this region are completely aborted by the IIO.
12:1	RV	0h	Reserved.
0	RW-LB	0b	vtd_chipset_base_address_enable Intel® VT-d base address enable Accesses to registers pointed to by VTBAR are accessible via the message channel or JTAG mini-port, irrespective of the setting of this enable bit; that is, even if this bit is clear, read and write to Intel® VT-d registers are completed normally (writes update registers and reads return the value of the register) for accesses from the message channel or JTAG mini-port. This bit is RW-LB (that is, the lock is determined based on the "trusted" bit in the message channel) when VTGENCTRL[15] is set; otherwise, it is RO.

7.7 DMIRCBAR

Base address register for DMI.

Bus: RootBus0 Device: 0Function: 0Offset: 50h Mode: DMI			
Bit	**Attribute**	**Default**	**Description**
31:12	RW-LB	00000h	DMIRCBAR This field corresponds to bits 32 to 12 of the base address DMI Root Complex register space. BIOS will program this register resulting in a base address for a 4KB block of contiguous memory address space. This register ensures that a naturally aligned 4KB space is allocated within the first 64GB of addressable memory space. System Software uses this base address to program the DMI Root Complex register set. All the Bits in this register are locked in LT mode. Note that this register is kept around on Device#0 even if that port is operating as PCIe port, to provide flexibility of using the VCs in PCIe mode as well. Nobody is asking for this capability right now but maintaining that flexibility.
11:1	RV	000h	Reserved
0	RW-LB	0b	DMIRCBAREN **0:** DMIRCBAR is disabled and does not claim any memory **1:** DMIRCBAR memory mapped accesses are claimed and decoded Notes: Accesses to registers pointed to by the DMIRCBAR, via message channel or JTAG mini-port are not gated by this enable bit i.e. accesses these registers are honored regardless of the setting of this bit. BIOS sets this bit only when it wishes to update the registers in the DMIRCBAR. It must clear this bit when it has finished changing values. This is required to ensure that the registers cannot be changed during an LT lock. This bit is protected by LT mode, but the registers in DMIRCBAR are not protected except by this bit.

7.8 PERFCTRLSTS_0: Performance Control and Status

Bus: RootBus	Device: 0-3	Function: 0	Offset: 180h
Bit	**Attribute**	**Default**	**Description**
31:21	RV	0	Reserved
20:16	RW	18h	**outstanding_requests_gen1** Number of outstanding RFOs and non-posted requests from a given PCIe* port. This register controls the number of outstanding inbound non-posted requests - I/O, configuration, memory - (maximum length of these requests is a single 64B cacheline) that a PCIe* 1.0 downstream port can have. This register provides the value for the port when it is operating in 1.0 mode and for a link width of x4. The BIOS programs this register based on the read latency to the main memory. This register also specifies the number of RFOs that can be kept outstanding on IDI for a given port. The link speed of the port can change during a PCIe* hotplug event and the port must use the appropriate multiplier. A value of "1" indicates one outstanding pre-allocated request, "2" indicates two outstanding pre-allocated requests, and so on. If the software programs a value greater than the buffer size, the DMA engine supports, then the maximum hardware supported value is used. The current BIOS recommendation is to leave this field at its default value.
15:14	RV	0	Reserved.
13:8	RW	30h	**outstanding_requests_gen2** Number of outstanding RFOs and non-posted requests from a given PCIe* port. This register controls the number of outstanding inbound non-posted requests - I/O, configuration, memory - (maximum length of these requests is a single 64B cacheline) that a PCIe* 2.0 downstream port can have. This register provides the value for the port when it is operating in 2.0 mode and for a link width of x4. The BIOS programs this register based on the read latency to the main memory. This register also specifies the number of RFOs that can be kept outstanding on IDI for a given port. The link speed of the port can change during a PCIe* hotplug event, and the port must use the appropriate multiplier. A value of "1" indicates one outstanding pre-allocated request, "2" indicates two outstanding pre-allocated requests, and so on. If the software programs a value greater than the buffer size the DMA engine supports, then the maximum hardware supported value is used. The current BIOS recommendation is to leave this field at its default value.
7	RW	1b	**Use_Allocating_Flow_Wr** Forcing all snooping writes from this port will use the allocating flow. **1:** All snooping writes received on this port use the allocating flow. **0:** All snooping writes will use the non-allocating flow. ***Note:*** VC1/VCm traffic is not impacted by this bit since all writes from VC1 and VCm are always non-snoop.
6	RV	0b	Reserved.

Bus: RootBus	**Device: 0-3**	**Function: 0**	**Offset: 180h**
Bit	**Attribute**	**Default**	**Description**
5	RW	0b	**ForceNoSnoopWrEn** Force no-snoop on VC0 writes received on this port **1:** All writes received on this port are treated as though the NS bit is set and will use the non-snoop non-allocating flow. **0:** Writes will be treated as non-snoop only if the NS bit is set and the NoSnoopWrEn bit is set. **Notes:** • VC1/VCm traffic is not impacted by this bit since all writes from VC1 and VCm are always non-snoop • This bit has precedence over the NoSnoopWrEn field in this register • Forcing writes to be non-snoop with this bit takes precedence over TPH
4	RW	1b	**read_stream_interleave_size**
3	RW	0b	**NoSnoopOpWrEn** Enable no-snoop on VC0 writes received on this port This applies to writes with the following conditions: NS=1 AND (TPH=0 OR TPHDIS=1) **1:** Inbound writes to memory with above conditions will be treated as non-coherent (no snoops) writes - IIO will use the non-allocating NS flow **0:** Inbound writes to memory with the previous conditions will be treated as allocating writes **Notes:** • If TPH=1 and TPHDIS=0 then NS is ignored, and this bit is ignored • VC1 and VCm writes are not controlled by this bit since they are always non-snoop and can be no other way • TPH and ForceNoSnoopWrEn have higher precedence than this bit
2	RW	0b	**NoSnoopOpRdEn** Enable no-snoop on VC0 reads
1	RW	0b	**read_passing_read_disable** Disable reads bypassing other reads
0	RW	1b	**read_stream_policy**