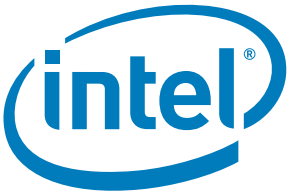
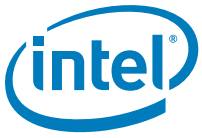


Second Generation Intel® Xeon® Scalable Processors

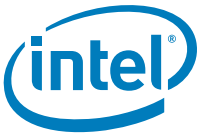
Datasheet, Volume Two: Registers

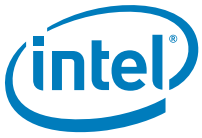
April 2019

Reference Number: 338846-001US



Contents



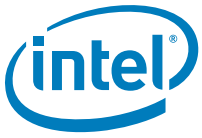


Tables

Revision History

Document Number	Revision Number	Description	Date
338846	001	• Initial Release	April 2019

§



1 Introduction

The Datasheet Volume 2 provides configuration space registers (CSRs).

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

- Second Generation Intel® Xeon® Bronze 3XXX processor
- Second Generation Intel® Xeon® Silver 4XXX processor
- Second Generation Intel® Xeon® Gold 5XXX processor
- Second Generation Intel® Xeon® Gold 6XXX processor
- Second Generation Intel® Xeon® Platinum 8XXX processor

The Second Generation Intel® Xeon® Scalable Processors is the next generation of 64-bit, multi-core server processor built on 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 Processors and the Platform Controller Hub (PCH). Included in this family of processors are integrated memory controller (IMC) and an Integrated I/O (IIO) on a single silicon die.

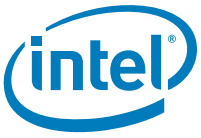
All processor types support up to 48 lanes of PCI Express* 3.0 links capable of 8.0 GT/s, and 4 lanes of DMI3/PCI Express 3.0. It features 2 Integrated Memory Controllers (IMC), each IMC supports up to three DDR4 channels with up to 2 DIMMs per channel.

Note: For supported processor configurations refer to: Second Generation Intel® Xeon® Scalable Processors Datasheet: Volume 1- Electrical, 338845.

1.1 Registers Overview and Configuration Process

This is volume two (Vol 2) of the processor public document, which provides uncore register and core MSR information for the processor. This volume documents the Configuration Space Registers (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.



Note: Some Default values will vary based on processor type and SKU, and in most cases these are the read only register fields which provide processor support visibility to firmware. Firmware should not rely on these Default values provided in this document, and instead verify these values by reading them with firmware.

1.2 Related Publications

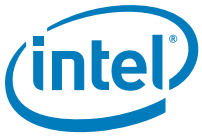
Refer to the following documents for additional information.

Table 1-1. Related Publications

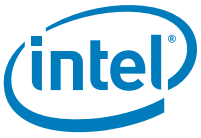
Document	Document Number / Location
• Second Generation Intel® Xeon® Scalable Processors Datasheet: Volume 1 - Electrical	338845
• Second Generation Intel® Xeon® Scalable Processors Specification Update	338848
• Second Generation Intel® Xeon® Scalable Processors Thermal Mechanical Design Guidelines	338847
• Intel® C620 Series Chipset Datasheet	336067
• Intel® C620 Series Chipset Thermal Mechanical Design Guidelines	336068
Intel®64 and IA-32Architectures Software Developer's Manuals *Volume 1: Basic Architecture* *Volume 2A: Instruction Set Reference, A-M* Volume 2B: Instruction Set Reference, N-Z Volume 3A: System Programming Guide Volume 3B: System Programming Guide Intel® 64 and IA-32Architectures Optimization Reference Manual	325462 http://www.intel.com/products/processor/manuals/index.htm
Intel® Virtualization Technology Specification for Directed I/O Architecture Specification	http://www.intel.com/content/www/us/en/intelligent-systems/intel-technology/vt-directed-io-spec.html
Intel®Trusted Execution Technology Software Development Guide	http://www.intel.com/technology/security/

1.2.1 Terminology

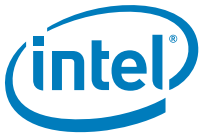
Term	Description
AC	Read and Write Access Control
ASPM	Active State Power Management
Intel AVX	Intel Advanced Vector Extensions (AVX) promotes legacy 128-bit SIMD instruction sets that operate on XMM register set to use a "vector extension" (VEX) prefix and operates on 256-bit vector registers (YMM).
Intel AVX 512	The base of the 512-bit SIMD instruction extensions are referred to as Intel® AVX-512 foundation instructions. They include extensions of the 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.
BMC	Baseboard Management Controller



Term	Description
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 (Coherency Agent) and HA (Home Agent).
CP	Control Policy
DDR4	Fourth generation Double Data Rate SDRAM memory technology.
DMA	Direct Memory Access
DMI3	Direct Media Interface Gen3 operating at PCI Express 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 memory to be marked as executable or non-executable, when combined with a supporting operating system. If code attempts to run in non-executable memory the processor raises an error to the operating system. This feature can prevent some classes of viruses or worms that exploit buffer overrun vulnerabilities and can thus help improve the overall security of the system. See the Intel® 64 and IA-32 Architectures Software Developer's Manuals for more detailed information.
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 SSE/SSE2 (Streaming SIMD Extensions) floating point instruction set to 256b operands.
HA	A Home Agent (HA) orders 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.
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 (Intel® UPI)	A cache-coherent, link-based Interconnect specification for Intel processors. Also known as Intel UPI.
Intel® 64 Technology	64-bit memory extensions to the IA-32 architecture. Further details on Intel 64 architecture and programming model can be found at http://developer.intel.com/technology/intel64/



Term	Description
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 (Intel® VT)	Processor Virtualization which 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 (IVR): The processor supports several integrated voltage regulators.
Intel UPI	Intel® Ultra Path Interconnect (Intel® UPI) Agent. An internal logic block providing interface between internal mesh and external Intel UPI.
LLC	Last Level Cache
LRDIMM	Load Reduced Dual In-line Memory Module
LRU	Least Recently Used. A term used in conjunction with cache allocation policy.
M2M	Mesh to Memory. Logic in the IMC which interfaces the IMC to the mesh.
M2PCIe	The logic in the IIO modules which interface the modules to the mesh.
MCP	A module in the IIO enabled in Ice Lake Server with Fabric which is used to interface to the on package Intel® Omni-Path.
MESH	The on die interconnect which connects modules in the processor.
MESI	Modified/Exclusive/Shared/Invalid. States used in conjunction with cache coherency
MLC	Mid Level Cache
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 \ NodeID	Node ID (NID) or NodeID (NID). The processor implements up to 4-bits of NodeID (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.
PCU	Power Control Unit.



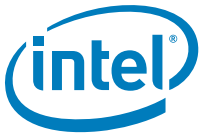
Term	Description
PCI Express 3.0	The third generation PCI Express specification that operates at twice the speed of PCI Express 2.0 (8 Gb/s); PCI Express 3.0 is completely backward compatible with PCI Express 1.0 and 2.0.
PCI Express 2.0	PCI Express Generation 2.0
PECI	Platform Environment Control Interface
Phit	The data transfer unit on Intel UPI at the Physical layer is called a Phit (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 which can contain multiple execution cores. Each execution core has an instruction cache and 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.
RDIMM \ LRDIMM	Registered Dual In-line Memory Module \ Load Reduced 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	Stock Keeping Unit (SKU) is a subset of a processor type with specific features, electrical, power and thermal specifications. Not all features are supported on all SKUs. A SKU is based on specific use condition assumption.
SSE	Intel® Streaming SIMD Extensions (Intel® SSE)
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. 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 packaging material) the processor must be handled in accordance with moisture sensitivity labeling (MSL) as indicated on the packaging material.
TAC	Thermal Averaging Constant
TDP	Thermal Design Power
TSOD	Temperature Sensor On DIMM
UDIMM	Unbuffered Dual In-line Memory Module
Uncore	The portion of the processor comprising the shared LLC cache, CHA, IMC, PCU, Ubox, 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

Term	Description
Volume Management Device (VMD)	Volume Management Device (VMD) is a new technology used to improve PCIe management. VMD maps the PCIe* configuration space for child devices/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	IO 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 Lane(s)

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 volume two (Vol 2) of the processor datasheet document which provides the Configuration Space Registers (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 contained 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.

Note: Some Default values will vary based on processor type and SKU, and in most cases these are the read only register fields which provide processor support visibility to firmware. Firmware should not rely on these Default values provided in this document, and instead verify these values by reading them with firmware.

Note: There are 2 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 Configuration Registers (CSRs): CSRs are chipset specific registers that are located at 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 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 resurfaces VMD as an additional PCI segment, allowing child devices behind VMD to be visible via standard methods.
- Memory-mapped I/O 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/WriteMSR instructions 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 Message Channel through the Ubox but might 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 PCI Express 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 Second Generation Intel® Xeon® Processor E5 v4 product family is specified in the CPUBUSNO register which exists in the I/O module's configuration space. Bus number is derived by the max 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 BIOS depending on which socket is used. The specific bus number for all PCIe devices in the Second Generation Intel® Xeon® Processor E5 v4 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 Bus Number, Device Number and Function Number. Device configuration is based on the PCI Type 0 configuration conventions. All processor registers appear on the PCI bus assigned for the processor socket. Bus number is derived by the max bus range setting and processor socket number.

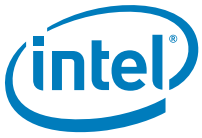
2.1.5 Unimplemented Devices/Functions and Registers

- Configuration reads to unimplemented functions and devices will return all ones emulating a master abort response. Note that there is no asynchronous error reporting that happens when a configuration read master aborts. Configuration writes to unimplemented functions and devices will return a normal response.
- 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 register (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/WriteMSR instructions. MSRs are always accessed as a naturally aligned 4 or 8 byte quantity.



For common IA-32 architectural MSRs, please refer to the *Intel® 64 and IA-32 Software Developer's Manual*.

2.1.7 Memory-Mapped I/O Registers

The PCI standard provides not only configuration space registers but also registers which reside in 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 are 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 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 2)

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 software.
RC	Read Clear Variant: These bits can be read by software, and the act of reading them automatically clears them. 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. Microcode that wants to ensure this bit was written must read wherever the side- effect takes place.
RW-O	Read / Write Once: These bits can be read by software. After reset, these bits can only be written by software once, after which the bits becomes 'Read Only'.
RW-L	Read / Write Lock: These bits can be read and written by 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 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 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 software. Writing a '0' to a bit clears it while writing a '1' has no effect.
ROS	RO Sticky: These bits can only be read by software, writes have no effect. The value of the bits is determined by the hardware only. These bits are only re-initialized 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 software to set a bit to "0". The 1->0 transition can only be performed by hardware. These registers are implicitly - V.
RWS	R / W Sticky: These bits can be read and written by software. These bits are only re-initialized to their default value by a PWRGOOD reset.
RW1CS	R / W1C Sticky: These bits can be read and cleared by software. Writing a '1' to a bit clears it, while writing a '0' to a bit has no effect. These bits are only re-initialized to their default value by a PWRGOOD reset.

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

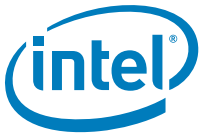
Attribute	Description
RW-LB	Read/Write Lock Bypass: Similar to RWL, these bits can be read and written by software. HW can make these bits "Read Only" via a separate configuration bit or other logic. However, RW-LB is a special case where the locking is controlled by the lock-bypass capability that is controlled by the lock-bypass enable bits. Each lock-bypass enable bit enables a set of config request sources that can bypass the lock. The requests sourced from the corresponding bypass enable bits will be lock-bypassed (i.e. 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. Note that the lock bits may not be sticky, and it is important that they are written to after reset to guarantee that 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 hardware. 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 software and may be modified by hardware. Software cannot expect the values to stay unchanged. These bits are re-initialized 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 software and may be modified by hardware. Software cannot expect the values to stay unchanged. These bits are re-initialized 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 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 software inside or outside of SMM but can only be written by 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 software while in SMM. Attempting to write these bits outside of SMM will cause a #GP exception to be raised.
SMM-RW1C	Read / Write 1 to Clear in SMM: These bits can be read and cleared by 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 software. When writing these bits, software must preserve the value read.
RSVD-Z	Reserved - Don't Care: These bits are reserved for future expansion and modifying their value has no effect. 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 1s and 0s, 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 Integrated Memory Controller (iMC) Configuration Registers

The Integrated Memory Controller registers are listed below and are specific to

- The Second Generation Intel® Xeon® Processor scalable family implement 2 Memory Controllers each with 3 DDR4 memory channels, 2 DIMMs per channel.
  - — The IMC Registers are implemented in the following Bus, Device, 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 12 Functions 0-5 for offsets >= 256, PCIe extended configuration space are not designed for direct usage by OS or device drivers, and may not be accessible directly by OS components such as device drivers. The PCI Capability Pointer Register (CAPPTR) is set to a value of 40h. BIOS/firmware and/or BMC can access these registers, combine the information obtained with system implementation specifics, and if required, make it available to the OS through firmware and/or BMC interfaces.

3.1 Device: 10,12 Function 0

3.1.1 pxpcap

PCI Express Capability.

Type: CFG Bus: 2 Device: 10, 12 Offset: 0x40			**PortID: N/A Function: 0**
Bit	**Attr**	**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): PCI Express Capability is Compliant with Version 1.0 of the PCI Express Spec. ***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.

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

3.1.2 mcmtr

Memory Technology

Type: CFG Bus: 2 Offset: 0x87c			PortID: N/A Device: 10, 12	Function: 0
Bit	**Attr**	**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 Read-Only (RO) with 0 value, that is, the directory is disabled. When this bit is set to zero, IMC ECC code 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, MC 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 re-initialize 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 8 TAD ranges (N + P + 1 = number of TAD ranges; P = how many times channel interleave changes within the SAD ranges.).

Note for mirroring configuration:

For 1-way interleave, channel 0-2 mirror pair: target list <0,2,x,x>, TAD ways = "00"

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

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

For 1-way interleave, lockstep mirroring, target list <0,2,x,x>, TAD ways = "00"

Type: CFG **PortID:** N/A **Bus:** 2 **Device:** 10, 12 **Function:** 0 **Offset:** 0x80, 0x84, 0x88, 0x8c, 0x90, 0x94, 0x98, 0x9c, 0xa0, 0xa4, 0xa8, 0xac			
Bit	**Attr**	**Default**	**Description**
31:12	RW_LB	0x0	TAD_LIMIT (tad_limit): Highest address of the range in system address space, 64MB granularity, i.e. TADRANGLIMIT[45:26].
11:10	RW_LB	0x0	Reserved TAD_SKT_WAY (tad_skt_way): socket interleave wayness 00 = 1 way, 01 = 2 way, 10 = 4 way, 11 = 8 way.
9:8	RW_LB	0x0	TAD_CH_WAY (tad_ch_way): Channel interleave wayness 00 - interleave across 1 channel or mirror pair 01 - interleave across 2 channels or mirror pairs 10 - interleave across 3 channels 11 - interleave across 4 channels This parameter effectively tells 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 HA, it tells how may channels to distribute the read requests across. When interleaving across 1 pair, this distributes the reads to two channels, when interleaving across 2 pairs, this distributes the reads across 4 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: CFG	PortID: N/A	
Bus: 2	Device: 10, 12	Function: 0
Offset: 0x8b4		

Bit	Attr	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 MC-reset. If it is clear when 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 need 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 doesn't 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 ad 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 IO starting to run RCOMP evaluation until RCOMP results are definitely ready. This counter is added in order to keep determinism of the process if operated in different mode. This register also indicates that first RCOMP has been done - required by BIOS.

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

Bit	Attr	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 be 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.
20:20	RW	0x0	no_mdll_fsm_override: Do not force DDRIO MDLL on during rcomp when set.

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

Bit	Attr	Default	Description
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

Capture externally asserted MEM_HOT[1:0]# assertion detection.

Type: CFG	PortID: N/A	
Bus: 2	Device: 10, 12	Function: 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 updated 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 updated 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/I2C* SCL and SDA signals that is used to access the Serial Presence Detect EEPROM (SPD) or Thermal Sensor on DIMM (TSOD) that defines the technology, configuration, and speed of the DIMMs controlled by iMC.

Type: CFG	PortID: N/A	
Bus: 2	Device: 10, 12	Function: 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 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.

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

Bit	Attr	Default	Description
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.
28:28	ROS_V	0x0	SMB_BUSY (smb_busy): SMBus Busy state. This bit is set by iMC while an SMBus/I2C command (including TSOD command issued from IMC hardware) is executing. Any transaction that is completed normally or gracefully will clear this bit automatically. By setting the SMB_SOFT_RST will also clear this bit. This register bit is sticky across reset so any surprise reset during pending SMBus operation will sustain the bit assertion across surprised 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 and/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 CHN and DIMM slot mapping: Slave Address: 0 -- Channel: Even Chn; Slot #: 0 Slave Address: 1 -- Channel: Even Chn; Slot #: 1 Slave Address: 2 -- Channel: Even Chn; Slot #: 2 Slave Address: 3 -- Channel: Even Chn; Slot #: 3 (reserved) Slave Address: 4 -- Channel: Odd Chn; Slot #: 0 Slave Address: 5 -- Channel: Odd Chn; Slot #: 1 Slave Address: 6 -- Channel: Odd Chn; Slot #: 2 Slave Address: 7 -- Channel: Odd Chn; Slot #: 3 (reserved) Since 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 DataHolds data read from SMBus Read commands. Since TSOD/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] = dont 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, please use byte access.

3.1.8 smbcmd_[0:1]

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

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

Bit	Attr	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/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. ***Important Note***: CPU hardware-based TSOD polling can be configured with pointer based access. If software manually issue SMBus transaction to other address, i.e. changing the pointer in the slave device, it is 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/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/I2C pointer update (byte). bit 30, and 29 are ignored. Note: SMBCntl_[0:1] [26] will NOT disable WrtPntr update command.
27:27	RWS	0x0	SMB_WRT_CMD (smb_wrt_cmd): When '0', it's a read command When '1', it's a write command
26:24	RWS	0x0	SMB_SA (smb_sa): Slave Address: This field identifies the DIMM SPD/TSOD to be accessed.
23:16	RWS	0x0	SMB_BA (smb_ba): Bus Txn 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 specified 1B access address.
15:0	RWS	0x0	SMB_WDATA (smb_wdata): Write Data: Holds data to be written by SPDW commands. Since TSOD/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] = dont care; SMB_WDATA[7:0] = writebyte. If we have a SMB slave 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 software responsibility to figure out the byte order of the slave access.

3.1.9 smbcntl_[0:1]

SMBus Control.

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

Bit	Attr	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 EEPROM's. '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***: Software need to set the SMB_CKOVRD back to 1 after 35ms in order 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 cleared the SMB_SBE. iMC added SMBus time-out control timer in B0. When the time-out control timer expired, the SMBCKOVRD# will "de-assert", i.e. 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, i.e. disabling SMBus write. This bit can only be written in SMMode. SMBus Read is not affected. I2C Write Pointer Update Command is not affected. ***Important Note to BIOS***: Since BIOS is the source to update SMBCNTL_x register initially after reset, it is important to determine whether the SMBus can have write capability before writing any upper bits (bit24-31) via byte-enable config write (or writing any bit within this register via 32b config write) within the SMBCNTL register.
10:10	RW	0x0	SMB_SOFT_RST (smb_soft_rst): 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 slave device is hung, software can write this bit to 1 and the SMB_CKOVRD to 0 (for more than 35ms)to force hung the SMB slaves to time-out and put it in idle state without using power good reset or warm reset. ***Note***: Software need to set the SMB_CKOVRD back to 1 after 35ms in order 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 cleared the SMB_SBE. If the IMC HW perform SMB time-out with the SMB_SBE_EN = 1. 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 graceful txn termination. Hardware will automatically de-assert the SMB_CKOVRD update to 1 after the pre-configured 35ms/65ms time-out.

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

Bit	Attr	Default	Description
9:9	RW_LB	0x0	start_tsod_poll: This bit starts the reading of all enabled devices. Note that 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 must be disabled (and pending TSOD polling must be drained) before changing the TSOD_POLL_EN.
7:0	RW_LB	0x0	TSOD_PRESENT for the lower and upper channels (tsod_present): DIMM slot mask to indicate whether the DIMM is equipped with 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: CFG	PortID: N/A	
Bus: 2	Device: 10, 12	Function: 0
Offset: 0xe8c, 0xe9c		

Bit	Attr	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, SMB_TSOD_POLL_RATE value is loaded into this counter and appear the updated value in the next DCLK.

3.1.11 smb_period_cfg

SMBus Clock Period Config.

Type: CFG	PortID: N/A	
Bus: 2	Device: 10, 12	Function: 0
Offset: 0xea0		

Bit	Attr	Default	Description
31:16	RWS	0x445c	Reserved
15:0	RWS	0xfa0	SMB_CLK_PRD (smb_clk_prd): This field specifies both SMBus Clock in number of DCLK. Note: In order 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 recommend to program an even value in this field since the hardware is simply doing a right shift for the divided by 2 operation. Note 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: CFG	PortID: N/A	Function: 0
Bus: 2	Device: 10, 12	
Offset: 0xea4		

Bit	Attr	Default	Description
31:16	RO_V	0x0	SMB1_CLK_PRD_CNTR (smb1_clk_prd_cntr): SMBus #1 Clock Period Counter for Ch 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 Ch 01. This field is the current SMBus Clock Period Counter Value.

3.1.13 smb_tsod_poll_rate

Type: CFG	PortID: N/A	Function: 0
Bus: 2	Device: 10, 12	
Offset: 0x1a8		

Bit	Attr	Default	Description
17:0	RWS	0x3e800	SMB_TSOD_POLL_RATE (smb_tsod_poll_rate): TSOD poll rate configuration between 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: CFG	PortID: N/A	Function: 1
Bus: 2	Device: 10, 12	
Offset: 0x40		

Bit	Attr	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/Port Type (device_port_type): Device type is Root Complex Integrated Endpoint
19:16	RO	0x1	Capability Version (capability_version): PCI Express Capability is Compliant with Version 1.0 of the PCI Express Spec. ***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.

Type: CFG Bus: 2 Offset: 0x40		PortID: N/A Device: 10, 12	Function: 1
Bit	**Attr**	**Default**	**Description**
7:0	RO	0x10	Capability ID (capability_id): Provides the PCI Express 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: CFG Bus: 2 Offset: 0x900		PortID: N/A Device: 10, 12	Function: 1
Bit	**Attr**	**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: CFG Bus: 2 Offset: 0x904		PortID: N/A Device: 10, 12	Function: 1
Bit	**Attr**	**Default**	**Description**
29:29	RW_LB	0x0	DisWPQWM (diswpqwm): Disable WPQ level based water mark, so that 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 max (# 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 SPAREINTERVAL.NORMOPDUR timer expiration.
28:24	RW_LB	0x0	HaFifoWM (hafifowm): minimum water mark for HA writes to failed rank. Actual wm is max of WPQ credit level and HaFifoWM. When wm is hit the HA is backpressured 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 max (# 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.

Type: CFG	PortID: N/A		
Bus: 2	Device: 10, 12	Function: 1	
Offset: 0x904			
Bit	**Attr**	**Default**	**Description**
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. Hardware clear after the sparing completion.

3.1.17 ssrstatus

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

Type: CFG	PortID: N/A		
Bus: 2	Device: 10, 12	Function: 1	
Offset: 0x94			
Bit	**Attr**	**Default**	**Description**
2:2	RW1C	0x0	PATCMPLT (patcmplt): All memory has been scrubbed. Hardware sets this bit each time the patrol engine steps through all memory locations. If software wants to monitor 0 --> 1 transition after the bit has been set, the software will need to clear the bit by writing a one to clear this bit in order 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 hardware once operation is complete. Bit is cleared by hardware when a new operation is enabled. Note: just before MC release the HA block prior to the completion of the sparing operation, 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 hardware once operation has started. It is cleared once 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 memtest, the failing address is logged in this register on memtest errors. 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: CFG	PortID: N/A	
Bus: 2	Device: 10, 12	Function: 1
Offset: 0x90C		

Bit	Attr	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. 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: CFG	PortID: N/A	
Bus: 2	Device: 10, 12	Function: 1
Offset: 0x910		

Bit	Attr	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 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: CFG	PortID: N/A	
Bus: 2	Device: 10, 12	Function: 1
Offset: 0x914		

Bit	Attr	Default	Description
31:31	RW_L	0x0	Scrub Enable (scrub_en): Scrub Enable when set.

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

Bit	Attr	Default	Description
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, patrol will stop at the end of the address range and set this bit. Patrol will resume from beginning of address range when this bit or stop_on_cmpl is cleared by BIOS and 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, patrol will log error then stop. On uncorr, patrol will convert to poison if enabled then stop. This mode is meant to be used as part of memory migration flow. Intel SMI is signaled by default.
27:27	RW_LBV	0x0	patrol stopped (ptl_stopped): When stop_on_err is set, patrol will stop on error and set this bit. Patrol will resume at the next address when this bit or stop_on_err is cleared by BIOS and 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.
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: Can only be changed when 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: CFG	PortID: N/A	
Bus: 2	Device: 10, 12	Function: 1
Offset: 0x91c		

Bit	Attr	Default	Description
28:16	RW-LB	0x320	NUMSPARE (numspare): Sparing operation duration. 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. System requests will be serviced during this interval.

3.1.22 rasenables

RAS Enables Register

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

Bit	Attr	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 iMC operation. This changes the error policy.

3.1.23 smisparectl

System Management Interrupt and Spare control register.

Type: CFG	PortID: N/A	
Bus: 2	Device: 10, 12	Function: 1
Offset: 0xb4		

Bit	Attr	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): (CMCI used as a proxy for 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: CFG	PortID: N/A	
Bus: 2	Device: 10, 12	Function: 1
Offset: 0x928		

Bit	Attr	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 threshold. The value of this field specify the bit position of the mask: 00h: reserved 01h: LEAKY_BUCKET_CNTR_LO bit 1, i.e. bit 12 of the full 53b counter ... 1Fh: LEAKY_BUCKET_CNTR_LO bit 31, i.e. bit 42 of the full 53b counter 20h: LEAKY_BUCKET_CNTR_HI bit 0, i.e. bit 43 of the full 53b counter ... 29h: LEAKY_BUCKET_CNTR_HI bit 9, i.e. 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 which is used by a 2-bit LEAKY_BKT_2ND_CNTR. LEAKY_BKT_2ND_CNTR_LIMIT specifies the value to generate LEAK pulse which is used to decrement the correctable error counter by 1 as shown below: 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 zeros in LEAKY_BUCKET_CFG register is equivalent to no leaky bucketing. BIOS must program this register to any non-zero value before switching to NORMAL mode.

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

Bit	Attr	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, i.e. bit 12 of the full 53b counter ... 1Fh: LEAKY_BUCKET_CNTR_LO bit 31, i.e. bit 42 of the full 53b counter 20h: LEAKY_BUCKET_CNTR_HI bit 0, i.e. bit 43 of the full 53b counter ... 29h: LEAKY_BUCKET_CNTR_HI bit 9, i.e. 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 which is used by a 2-bit LEAKY_BKT_2ND_CNTR. LEAKY_BKT_2ND_CNTR_LIMIT specifies the value to generate LEAK pulse which is used to decrement the correctable error counter by 1 as shown below: 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 zeros in LEAKY_BUCKET_CFG register is equivalent to no leaky bucketing. MRC BIOS must program this register to any non-zero value before switching to NORMAL mode.

3.1.25 leaky_bucket_cntr_lo

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

Bit	Attr	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 is not captured in 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: CFG **PortID:** N/A
Bus: 2 **Device:** 10, 12 **Function:** 1
Offset: 0x934

Bit	Attr	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 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: CFG **PortID:** N/A
Bus: 2 **Device:** 10, 12 **Function:** 2,3,4,5
Offset: 0x40

Bit	Attr	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/Port Type (device_port_type): Device type is Root Complex Integrated Endpoint
19:16	RO	0x1	Capability Version (capability_version): PCI Express Capability is Compliant with Version 1.0 of the PCI Express Spec. ***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 PCI Express capability ID assigned by PCI-SIG.

3.2.2 pxpenhcap

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

Type: CFG	PortID: N/A	
Bus: 2	Device: 10, 12	Function: 2,3,4,5
Offset: 0x100		

Bit	Attr	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: CFG	PortID: N/A	
Bus: 2	Device: 10,11,12	Function: 2,6
Offset: 0x40		

Bit	Attr	Default	Description
7:0	RO	0x10	Capability ID (capability_id): Provides the PCI Express capability ID assigned by PCI-SIG.

3.3.2 chn_temp_cfg

Type: CFG	PortID: N/A	
Bus: 2	Device: 10,11,12	Function: 2,6
Offset: 0x108		

Bit	Attr	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 configuration write to DIMMTEMPSTAT_[0:3][7:0].
27:27	RW	0x1	Enables thermal bandwidth throttling limit (bw_limit_thrt_en):
23:16	RW	0x0	THRT_EXT (thrt_ext): Max number of throttled transactions to be issued during BWLIMITTF due to externally asserted MEMHOT#.

Type: CFG	PortID: N/A		
Bus: 2	Device: 10,11,12	Function: 2,6	
Offset: 0x108			
Bit	**Attr**	**Default**	**Description**
15:15	RW	0x0	THRT_ALLOW_ISOCH (thrt_allow_isoch): When this bit is zero, MC will lower CKE during Thermal Throttling, and ISOCH is blocked. When this bit is one, MC will NOT lower CKE during Thermal Throttling, and ISOCH will be allowed base on bandwidth throttling setting. However, setting this bit would mean more power consumption due to CKE is 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: CFG	PortID: N/A		
Bus: 2	Device: 10,11,12	Function: 2,6	
Offset: 0x10c			
Bit	**Attr**	**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: CFG	PortID: N/A		
Bus: 2	Device: 10,11,12	Function: 2,6	
Offset: 0x110, 0x114			
Bit	**Attr**	**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 sensor does not support positive-going threshold hysteresis
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 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. Valid range: 32 - 127 in degree (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. Valid range: 32 - 127 in degree (C). Others: reserved.

3.3.5 dimm_temp_th_[0:2]

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

Bit	Attr	Default	Description
26:24	RW-LB	0x0	TEMP_THRT_HYST (temp_thrt_hyst): Positive going Threshold Hysteresis Value. Set to 00h if 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 degree (C). Note: the default value is listed in decimal. 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 degree (C). ***Note***: The default value is listed in decimal. 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 andor THRTMID and initiate Interrupt (MEMHOT#). ***Note***: The default value is listed in decimal.valid range: 32 - 127 in degree (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, i.e. all activities (ACT, READ, WRITE) from all ranks within a DIMM are tracked together in one DIMM activity counter. These throttle limits for hi and crit are also used during scalable memory buffer thermal throttling.

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

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

3.3.7 dimm_temp_ev_ofst_[0:1]

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

Bit	Attr	Default	Description
31:24	RO	0x0	TEMP_AVG_INTRVL (temp_avg_intrvl): Temperature data is averaged over this period. At the end of averaging period (ms), 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 temp 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 in order to avoid data lost when DIMM temperature is above 85C but below 95C. ***Warning***: If the 2x refresh is disable with extended temperature range DIMM configuration, system cooling and power thermal throttling scheme must guarantee the DIMM temperature will not exceed 85C.
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
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: CFG	PortID: N/A	
Bus: 2	Device: 10,11,12	Function: 2,6
Offset: 0x150, 0x154		

Bit	Attr	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 assertion is going to trigger 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 configurable interrupt (Either MEMHOT# only or both Intel SMI and MEMHOT#) defined in bit 30 of CHN_TEMP_CFG.

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

Bit	Attr	Default	Description
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 assertion is going to trigger 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 assertion is going to trigger 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 assertion is going to trigger 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, i.e. configuration write to this byte is aborted. This byte is updated from internal logic from a 2:1 Mux which can be selected from either CLTT temperature or from the corresponding temperature registers output (PCODE_TEMP_OUTPUT) updated from pcode. The mux select is controlled by CLTT_OR_PCODE_TEMP_MUX_SEL defined in CHN_TEMP_CFG register. Valid range from 0 to 127 i.e. 0C to +127C. Any negative value read from TSOD is forced to 0. TSOD decimal point value is also truncated to integer value.

3.3.9 thrt_pwr_dimm_[0:1]

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

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

Bit	Attr	Default	Description
15:15	RW	0x1	THRT_PWR_EN (thrt_pwr_en): bit[15]: set to one to enable the power throttling for the DIMM.
11:0	RW	0xfff	Power Throttling Control (thrt_pwr): bit[11:0]: Max number of transactions (ACT, READ, WRITE) to be allowed (per DIMM) during the 1 micro-sec throttling timeframe per power throttling.

3.4 Device 10,12 Functions 3,7

3.4.1 correrrcnt_0

Per Rank corrected error counters.

Type: CFG us: 2 Offset: 0x104			PortID: N/A Device: 10,12	Function: 3,7
Bit	**Attr**	**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 BIOS.	
30:16	RWS_LV	0x0	RANK 1 CORRECTABLE ERROR COUNT (cor_err_cnt_1): The corrected error count for this rank. 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 BIOS.	
14:0	RWS_LV	0x0	RANK 0 CORRECTABLE ERROR COUNT (cor_err_cnt_0): The corrected error count for this rank. Hardware automatically clear 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: CFG us: 2 Offset: 0x108			PortID: N/A Device: 10,12	Function: 3,7
Bit	**Attr**	**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 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 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: CFG Bus: 1 Offset: 0x10c			PortID: N/A Device: 20,21,23	Function: 2,3
Bit	**Attr**	**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 BIOS.	
30:16	RWS_LV	0x0	RANK 5 COR_ERR_CNT (cor_err_cnt_5): The corrected error count for this rank.	

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

Bit	Attr	Default	Description
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 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: CFG	PortID: N/A	
Bus: 1	Device: 20,21,23	Function: 2,3
Offset: 0x110		

Bit	Attr	Default	Description
31:31	RW1CS	0x0	RANK 7 OVERFLOW (overflow_7): The corrected error count for this rank.
30:16	RWS_LV	0x0	RANK 7 COR_ERR_CNT_7 (cor_err_cnt_7): The corrected error count for this rank.
15:15	RW1CS	0x0	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 BIOS.
14:0	RWS_LV	0x0	RANK 6 COR_ERR_CNT (cor_err_cnt_6): The corrected error count for this rank.

3.4.5 correrrthrshld_0

This register holds the per rank corrected error thresholding value.

Type: CFG	PortID: N/A	
us: 2	Device: 10,12	Function: 3,7
Offset: 0x11c		

Bit	Attr	Default	Description
30:16	RW-LB	0x7fff	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.
14:0	RW-LB	0x7fff	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.

3.4.6 correrrthrshld_1

This register holds the per rank corrected error thresholding value.

Type: CFG us: 2 Offset: 0x120		PortID: N/A Device: 10,12	Function: 3,7
Bit	**Attr**	**Default**	**Description**
30:16	RW-LB	0x7fff	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.
14:0	RW-LB	0x7fff	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.

3.4.7 correrrthrshld_2

This register holds the per rank corrected error thresholding value.

Type: CFG us: 2 Offset: 0x124		PortID: N/A Device: 10,12	Function: 3,7
Bit	**Attr**	**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: CFG us: 2 Offset: 0x128		PortID: N/A Device: 10,12	Function: 3,7
Bit	**Attr**	**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 bios.

Type: CFG us: 2 Offset: 0x134		PortID: N/A Device: 10,12	Function: 3,7
Bit	**Attr**	**Default**	**Description**
31:24	RW_V	0x0	ddr4crc_rank_log: This field get set with 1'b1 if the corresponding rank detected DDR4 CRC in one of its write data. This will be cleared by 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: CFG us: 2 Offset: 0x138		PortID: N/A Device: 10,12	Function: 3,7
Bit	**Attr**	**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 secondary leaky bucket counter limit for all 8 logical ranks within 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 roll over at the predefined counter limit. The counter increment 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 roll over at 1. 2: the LEAK pulse is generated one DCLK after the counter roll over at 2. 3: the LEAK pulse is generated one DCLK after the counter roll over at 3.

Type: CFG us: 2 Offset: 0x138		PortID: N/A Device: 10,12	Function: 3,7
Bit	**Attr**	**Default**	**Description**
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), hardware

will generate a LINK interrupt and log (and preserve) the failing device in the FailDevice field. SMM software will read the failing device on the particular rank. Software then set the EN bit to enable substituion of the failing device/rank with the parity from the rest of the devices inline.

For independent channel configuration, each rank can tag once. Up to 8 ranks can be tagged.

For lock-step channel configuration, only one x8 device can be tagged per rank-pair. 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.

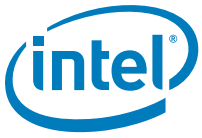
There is no hardware logic to report incorrect programming error. Unpredicable error and/or silent data corruption will be the consequence of such programming error.

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 would drop the correction capability and any subsequent ECC error from this rank would cause uncorrectable error.

Type: CFG us: 2 Offset: 0x140, 0x141, 0x142, 0x143, 0x144, 0x145, 0x146, 0x147		PortID: N/A Device: 10,12	Function: 3,7
Bit	**Attr**	**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. ECC error "detection" capability will not degrade after setting this bit. For lock-step channel configuration, only one x8 device can be tagged per rank-pair. 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 fail device. The DEVTAG_CNTL_x.EN on the other channel of the corresponding rank must not be set.

Type: CFG	PortID: N/A		
us: 2	Device: 10,12	Function: 3,7	
Offset: 0x140, 0x141, 0x142, 0x143, 0x144, 0x145, 0x146, 0x147			
Bit	**Attr**	**Default**	**Description**
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. ECC error "detection" capability will not degrade after setting this bit. Warning: For lock-step channel configuration, only one x8 device can be tagged per rank-pair. 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 fail 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, 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 HW on completion of DDDC sparing.

§



4 Intel UPI Registers

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)

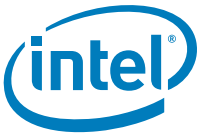
4.1 Bus: 3, Device: 16,14, Function: 3

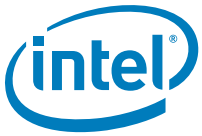
4.1.1 ktimiscstat

Intel UPI Misc Status

Bus: B(3)	Device: 16-14	Function: 3	Offset: D4
Bit	**Attr**	**Default**	**Description**
31:3	RSVD-Z	00000000h	Reserved — don't 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 above.

§





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 coordination for the performance architecture, and scratchpad and semaphore registers.

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

5.1.1 VID

PCI Vendor ID Register

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

5.1.2 DID

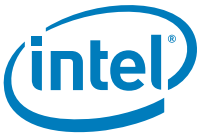
PCI Device Identification Number

Bus: B(0)	Device: 8	Function: 0	Offset: 2
Bit	**Attr**	**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	**Attr**	**Default**	**Description**
31:16	RSVD-Z	0000h	Reserved — don't 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 — don't care.
7:5	RW-LB	0h	**LockNodeId** — NodeId of the lock master
4:3	RSVD-Z	0h	Reserved — don't care.
2:0	RW-LB	0h	**LclNodeId** — Node Id of the local Socket



5.1.4 IntControl

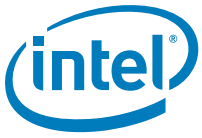
Interrupt Configuration Register

Bus: B(0)	Device: 8	Function: 0	Offset: C8
Bit	**Attr**	**Default**	**Description**
31:19	RSVD-Z	0000h	Reserved — don't care.
18	RW-LB	0h	**LogFlatClustOvrEn** — 0: IA32 Logical Flat or Cluster Mode bit is locked as Read only bit. 1: IA32 Logical Flat or Cluster Mode bit may be written by SW, values written by xTPR update are ignored. For one time override of the IA32 Logical Flat or Cluster Mode value, return this bit to it's default state after the bit is changed. Leaving this bit as '1' will prevent automatic update of the filter.
17	RW-LBV	0h	**LogFltClustMod** — Set by BIOS to indicate if the OS is running 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 flat/ cluster mode bit in the IntPrioUpd message as part of setting bit 1 in this register 1: Enable the above checking
15:11	RSVD-Z	00h	Reserved — don't 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 select bits 7:4/5:4 for vector cluster/flat algorithm 001 select bits 6:3/4:3 010 select bits 4:1/2:1 011 select bits 3:0/1:0 other - reserved
7	RSVD-Z	0h	Reserved — don't care.
6:4	RW	0h	**RdrModSel** — Selects the redirection mode used for MSI interrupts with lowest-priority delivery mode. The following schemes are used: 000: Fixed Priority - select the first enabled APIC in the cluster. 001: Redirect last - last vector selected (applicable only in extended mode) 010: Hash Vector - select the first enabled APIC in round robin manner starting form the hash of the vector number. default: Fixed Priority
3:2	RSVD-Z	0h	Reserved — don't 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 extended XAPIC configuration to be used. This bit can be written directly, and can also be updated using XTPR messages

5.1.5 GIDNIDMAP

Mapping between group id and nodeid

Bus: B(0)	Device: 8	Function: 0	Offset: D4
Bit	**Attr**	**Default**	**Description**
31:24	RSVD-Z	00h	Reserved — don't care.
23:21	RW-LB	0h	**NodeId7** — NodeId for group id 7
20:18	RW-LB	0h	**NodeId6** — Node Id for group 6
17:15	RW-LB	0h	**NodeId5** — Node Id for group 5
14:12	RW-LB	0h	**NodeId4** — Node Id for group id 4



Bus: B(0)	Device: 8	Function: 0	Offset: D4
Bit	**Attr**	**Default**	**Description**
11:9	RW-LB	0h	**NodeId3** — Node Id for group 3
8:6	RW-LB	0h	**NodeID2** — Node Id for group Id 2
5:3	RW-LB	0h	**NodeId1** — Node Id for group Id 1
2:0	RW-LB	0h	**NodeId0** — Node Id for group 0

5.1.6 UBOXErrSts

This is error status register in the Ubox and covers most of the interrupt related errors

Bus: B(0)	Device: 8	Function: 0	Offset: C8
Bit	**Attr**	**Default**	**Description**
31:24	RSVD-Z	00h	Reserved — don't care.
23:18	RWS-V	00h	**Msg_Ch_Tkr_TimeOut** — Message Channel Tracker TimeOut. This error occurs when any NP request doesn't receive response in 4K cycles.
17	RWS-V	0h	**Msg_Ch_Tkr_Err** — Message Channel Tracker Error. This error occurs such case that illegal broad cast 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 Ubox
6	RWS-V	0h	**SMITimeOut** — SMI Timeout received by Ubox
5	RWS-V	0h	**CFGWrAddrMisAligned** — MMCFG Write Address Misalignment received by Ubox. All MMCFG access must be less than or equal to 4B in length and cannot cross a 4B boundary. When Ubox sees a misaligned MMCFG access, it will be aborting the transaction.
4	RWS-V	0h	**CFGRdAddrMisAligned** — MMCFG Read Address Misalignment received by Ubox. All MMCFG access must be less than or equal to 4B in length and cannot cross a 4B boundary. When Ubox sees a misaligned MMCFG access, it will be aborting the transaction.
3	RWS-V	0h	**UnsupportedOpcode** — Unsupported opcode received by Ubox
2	RWS-V	0h	**PoisonRsvd** — 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	**Attr**	**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	**Attr**	**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	**Attr**	**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

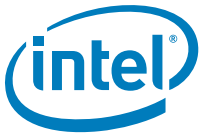
Bus: B(0)	Device: 8	Function: 2	Offset: D0
Bit	**Attr**	**Default**	**Description**
31:16	RSVD-Z	0000h	Reserved — don't care.
15:8	RW-LB	05h	**CPUBUSNO5** — Bus Number 5
7:0	RW-LB	04h	**CPUBUSNO4** — Bus Number 4

5.2.4 SMICtrl

SMI generation control

Bus: B(0)	Device: 8	Function: 2	Offset: D8
Bit	**Attr**	**Default**	**Description**
31:29	RSVD-Z	0h	Reserved — don't 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 — don't care.
19:0	RSVD-P	00000h	Reserved — protected.

§



6 Power Control Unit (PCU) Registers

The Power Control Unit (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	**Attr**	**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	**Attr**	**Default**	**Description**	
15:0	RO	2080h	Device_Identification_Number —	

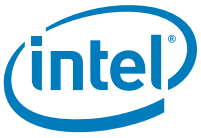
6.1.3 PACKAGE_ENERGY_STATUS

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	**Attr**	**Default**	**Description**	
31:0	RO-V	00000000h	**DATA** — Refer to MSR 611h which this is a mirror of for description.	

6.1.4 MEM_TRML_TEMPERATURE_REPORT_0

This register is used to report the thermal status of the memory. The channel max 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	**Attr**	**Default**	**Description**
31:24	RSVD-P	00h	Reserved — protected.
23:16	RO-V	00h	**Channel2_Max_Temperature** — Temperature in Degrees (C).
15:8	RO-V	00h	**Channel1_Max_Temperature** — Temperature in Degrees (C).
7:0	RO-V	00h	**Channel0_Max_Temperature** — Temperature in Degrees (C).

6.1.5 MEM_TRML_TEMPERATURE_REPORT_1

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

MEM_TRML_TEMPERATURE_REPORT_1 is used for channel temperature of DIMMs under IMC1

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

6.1.6 MEM_TRML_TEMPERATURE_REPORT_2

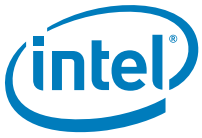
This register is used to report the thermal status of the memory. The channel max temperature field is used to report the maximal temperature of all ranks.

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

6.1.7 PACKAGE_TEMPERATURE

Package temperature in degrees (C). This field is updated by FW.

Bus: B(1)Device: 30Function: 0Offset: C8			
Bit	**Attr**	**Default**	**Description**
31:8	RSVD-Z	000000h	Reserved — don't care.
7:0	RO-V	00h	**DATA** — Package temperature in degrees (C).



6.1.8 TEMPERATURE_TARGET

Legacy register holding temperature related constants for Platform use.

Bus: B(1)Device: 30Function: 0Offset: E4			
Bit	**Attr**	**Default**	**Description**
31:28	RSVD-Z	0h	Reserved — don't care.
27:24	RW	0h	**TJ_MAX_TCC_OFFSET** — Refer to MSR 1A2h which this is a mirror of for description.
23:16	RO-V	00h	**REF_TEMP** — Refer to MSR 1A2h which this is a mirror of for description.
15:8	RO-V	00h	**FAN_TEMP_TARGET_OFST** — Refer to MSR 1A2h which this is a mirror of for description.
7:0	RSVD-Z	00h	Reserved — don't 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	**Attr**	**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	**Attr**	**Default**	**Description**	
15:0	RO	2082h	**Device_Identification_Number** —	

6.2.3 DRAM_ENERGY_STATUS

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 SW.

6.2.4 PACKAGE_RAPL_PERF_STATUS

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

This register is used to report Package Power limit violations.

6.2.5 DRAM_POWER_INFO

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

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

6.2.6 DRAM_RAPL_PERF_STATUS

This register is used by Pcode to report DRAM Plane Power limit violations in the Platform.

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

6.2.7 THERMTRIP_CONFIG

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

Bus: B(1)Device: 30Function: 2Offset: F8			
Bit	**Attr**	**Default**	**Description**
31:4	RSVD-Z	0000000h	Reserved — don't 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, PCU will ignore the MEMtrip information and ThermTrip will just have the processor indication. Expect BIOS to Enable this in Phase4

§