

# **Processor Programming Reference (PPR) for AMD Family 1Ah Model 24h, Revision B0 Processors**

# Legal Notices

© 2023,2024 Advanced Micro Devices, Inc. All rights reserved.

The information contained herein is for informational purposes only, and is subject to change without notice. While every precaution has been taken in the preparation of these materials, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of these materials, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by these materials. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale.

## Trademarks:

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

AGESA is a trademark of Advanced Micro Devices, Inc.

AMD Virtualization is a trademark of Advanced Micro Devices, Inc.

Adobe is a registered trademark of Adobe.

Arm is a registered trademark of Arm Limited.

CXL is a trademark of Compute Express Link Consortium, Inc.

DirectX is a registered trademark of Microsoft Corporation.

HDMI is a trademark of HDMI Licensing, LLC.

Infinity Fabric is a trademark of Advanced Micro Devices, Inc.

Linux is a registered trademark of Linus Torvalds.

MIPI I3C is a registered trademark of MIPI Alliance.

Microsoft is a registered trademark of Microsoft Corporation.

OpenCL is a trademark of Apple, Inc. used by permission by Khronos Group, Inc.

OpenGL is a registered trademark of Hewlett Packard Enterprise.

PCI Express is a registered trademark of PCI-SIG Corporation.

PCIe is a registered trademark of PCI-SIG Corporation.

Windows is a registered trademark of Microsoft Corporation.

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

Reverse engineering or disassembly is prohibited.

USE OF THIS PRODUCT IN ANY MANNER THAT COMPLIES WITH THE MPEG ACTUAL OR DE FACTO VIDEO AND/OR AUDIO STANDARDS IS EXPRESSLY PROHIBITED WITHOUT ALL NECESSARY LICENSES UNDER APPLICABLE PATENTS. SUCH LICENSES MAY BE ACQUIRED FROM VARIOUS THIRD PARTIES INCLUDING, BUT NOT LIMITED TO, IN THE MPEG PATENT PORTFOLIO, WHICH LICENSE IS AVAILABLE FROM MPEG LA, L.L.C., 6312 S. FIDDLERS GREEN CIRCLE, SUITE 400E, GREENWOOD VILLAGE, COLORADO 80111.

# List of Chapters

- 1 Overview**
- 2 Core Complex (CCX)**
- 3 Reliability, Availability, and Serviceability (RAS) Features**
- 4 Advanced Platform Management Link (APML)**
- 5 SB Temperature Sensor Interface (SB-TSI)**
- 6 DXIO**
- 7 Fusion Controller Hub (FCH)**

**List of Namespaces**

**List of Definitions**

# Table of Contents

<b>1</b>	<b>Overview</b>
1.1	Intended Audience
1.2	Reference Documents
1.2.1	Documentation Conventions
1.3	Adobe® Reader
1.3.1	Adobe® Reader Configuration
1.3.1.1	Open Hyperlink Document in New Window
1.3.2	Adobe® Reader Usage
1.3.2.1	Searching a Multiple Volume PPR
1.3.2.2	Cross-References and Hyperlinks
1.3.2.3	Find Current Bookmark
1.4	Conventions
1.4.1	Numbering
1.4.2	Arithmetic And Logical Operators
1.4.2.1	Operator Precedence and Associativity
1.4.3	Register Mnemonics
1.4.3.1	Logical Mnemonic
1.4.3.2	Physical Mnemonic
1.4.4	Register Format
1.4.4.1	Register Instances
1.4.4.2	Register Physical Mnemonic, Title, and Name
1.4.4.3	Full Width Register Attributes
1.4.4.4	Register Description
1.4.4.5	Register Instance Table
1.4.4.5.1	Content Ordering in a Row
1.4.4.5.2	Multiple Instances Per Row
1.4.4.5.3	MSR Access Method
1.4.4.5.3.1	MSR Per-Thread Example
1.4.4.5.3.2	MSR Range Example
1.4.4.5.4	BAR Access Method
1.4.4.5.4.1	BAR as a Register Reference
1.4.4.5.5	PCICFG Access Method
1.4.4.5.5.1	PCICFG Bus Implied to be 00h
1.4.4.5.6	Data Port Access Method
1.4.4.6	Register Field Format
1.4.4.7	Simple Register Field Format
1.4.4.8	Complex Register Field Format
1.4.4.9	Field Name is Reserved
1.4.4.10	Field Access Type
1.4.4.10.1	Conditional Access Type Expression
1.4.4.11	Field Reset
1.4.4.12	Field Initialization
1.4.4.13	Field Check
1.4.4.14	Field Valid Values
1.4.5	Revision History and Change Bar Notation
1.5	Definitions
1.6	Changes Between Revisions and Product Variations
1.6.1	Revision Conventions
1.7	Package
1.7.1	Package type



1.8	Processor Overview
1.8.1	Features
<b>2</b>	<b>Core Complex (CCX)</b>
2.1	Processor x86 Core
2.1.1	Core Functional Information
2.1.1.1	Core Definitions
2.1.2	Secure Virtual Machine Mode (SVM)
2.1.2.1	BIOS support for SVM Disable
2.1.2.1.1	Enable AMD Virtualization™
2.1.2.1.2	Disable AMD Virtualization™
2.1.2.1.3	Disable AMD Virtualization™, with a user supplied key
2.1.3	Microcode Patching
2.1.3.1	Microcode Patching Overview
2.1.3.2	Microcode Patch Block (MPB)
2.1.4	Effective Frequency
2.1.5	Address Space
2.1.5.1	Virtual Address Space
2.1.5.2	Physical Address Space
2.1.5.3	System Address Map
2.1.5.3.1	Memory Access to the Physical Address Space
2.1.5.3.1.1	Determining Memory Type
2.1.6	Configuration Space
2.1.6.1	Memory Mapped IO (MMIO) Configuration Coding Requirements
2.1.6.2	MMIO Configuration Ordering
2.1.6.3	Processor Configuration Space
2.1.7	PCI Configuration Legacy Access
2.1.8	System Software Interaction With SMT Enabled
2.1.9	Register Sharing
2.1.10	Timers
2.1.11	Interrupts
2.1.11.1	System Management Mode (SMM)
2.1.11.1.1	SMM Overview
2.1.11.1.2	Mode and Default Register Values
2.1.11.1.3	SMI Sources And Delivery
2.1.11.1.4	SMM Initial State
2.1.11.1.5	SMM Save State
2.1.11.1.6	System Management State
2.1.11.1.7	Exceptions and Interrupts in SMM
2.1.11.1.8	The Protected ASeg and TSeg Areas
2.1.11.1.9	SMM Special Cycles
2.1.11.1.10	Locking SMM
2.1.11.1.11	SMM Page Configuration Lock
2.1.11.2	Local APIC
2.1.11.2.1	Local APIC Functional Description
2.1.11.2.1.1	Detecting and Enabling
2.1.11.2.1.2	APIC Register Space
2.1.11.2.1.3	ApicId Enumeration Requirements
2.1.11.2.1.4	Physical Destination Mode
2.1.11.2.1.5	Logical Destination Mode
2.1.11.2.1.6	Interrupt Delivery
2.1.11.2.1.7	Vectored Interrupt Handling
2.1.11.2.1.8	Interrupt Masking
2.1.11.2.1.9	Spurious Interrupts

- 2.1.11.2.1.10 Spurious Interrupts Caused by Timer Tick Interrupt
- 2.1.11.2.1.11 Lowest-Priority Interrupt Arbitration
- 2.1.11.2.1.12 Inter-Processor Interrupts
- 2.1.11.2.1.13 APIC Timer Operation
- 2.1.11.2.1.14 Generalized Local Vector Table
- 2.1.11.2.1.15 State at Reset
- 2.1.11.2.2 Local APIC Registers
- 2.1.12 CPUID Instruction
- 2.1.12.1 CPUID Instruction Functions
- 2.1.13 MSR Registers
- 2.1.13.1 MSRs - MSR0000\_xxxx
- 2.1.13.2 MSRs - MSRC000\_xxxx
- 2.1.13.3 MSRs - MSRC001\_0xxx
- 2.1.13.4 MSRs - MSRC001\_1xxx
- 2.1.13.5 L3 MSRs - MSRC001\_1xxx
- 2.1.14 Performance Monitor Counters
- 2.1.14.1 RDPMC Assignments
- 2.1.14.2 Performance Measurement
- 2.1.14.3 Pipeline Utilization Analysis
- 2.1.14.4 Large Increment per Cycle Events
- 2.1.14.5 Core Performance Monitor Counters
- 2.1.14.5.1 Floating-Point (FP) Events
- 2.1.14.5.2 Load/Store (LS) Events
- 2.1.14.5.3 Instruction Cache (IC) and Branch Prediction (BP) Events
- 2.1.14.5.4 DE Events
- 2.1.14.5.5 EX (SC) Events
- 2.1.14.5.6 L2 Cache Events
- 2.1.14.6 L3 Cache Performance Monitor Counters
- 2.1.14.6.1 L3 Cache PMC Events
- 2.1.15 Instruction Based Sampling (IBS)
- 2.2 L3 Cache
- 2.2.1 L3 MSR Registers
- 2.2.2 L3 Clocks and Test (CT) MSR Registers.

### 3 Reliability, Availability, and Serviceability (RAS) Features

- 3.1 Machine Check Architecture
- 3.1.1 Overview
- 3.1.1.1 Legacy Machine Check Architecture
- 3.1.1.2 Machine Check Architecture Extensions
- 3.1.1.3 Use of MCA Information
- 3.1.1.3.1 Error Management
- 3.1.1.3.2 Fault Management
- 3.1.2 Machine Check Registers
- 3.1.2.1 Global Registers
- 3.1.2.2 Machine Check Banks
- 3.1.2.2.1 Legacy MCA Registers
- 3.1.2.2.2 Legacy MCA MSRs
- 3.1.2.2.3 MCAX Registers
- 3.1.2.2.4 MCAX MSRs
- 3.1.2.3 Access Permissions
- 3.1.3 Machine Check Errors
- 3.1.3.1 Error Severities
- 3.1.3.2 Exceptions and Interrupts
- 3.1.3.3 Error Codes

- 3.1.3.4 Extended Error Codes
- 3.1.3.5 DOER and SEER State
- 3.1.3.6 MCA Overflow Recovery
- 3.1.3.7 MCA Recovery
- 3.1.4 Machine Check Features
  - 3.1.4.1 Error Thresholding
  - 3.1.4.2 Error Simulation
- 3.1.5 Software Guidelines
  - 3.1.5.1 Recognizing MCAX Support
  - 3.1.5.2 Communicating MCAX Support
  - 3.1.5.3 Machine Check Initialization
  - 3.1.5.4 Determining Bank Count
  - 3.1.5.5 Determining Bank Type
  - 3.1.5.6 Recognizing Error Type
  - 3.1.5.7 Machine Check Error Handling
- 3.2 Machine Check Architecture Implementation
  - 3.2.1 Implemented Machine Check Banks
  - 3.2.2 Implemented Machine Check Bank Registers
  - 3.2.3 Mapping of Banks to Blocks
  - 3.2.4 Decoding Error Type
  - 3.2.5 MCA Banks
    - 3.2.5.1 LS
    - 3.2.5.2 IF
    - 3.2.5.3 L2
    - 3.2.5.4 DE
    - 3.2.5.5 EX
    - 3.2.5.6 FP
    - 3.2.5.7 L3
    - 3.2.5.8 CS
    - 3.2.5.9 PIE
    - 3.2.5.10 UMC
    - 3.2.5.11 NBIO
- 4 Advanced Platform Management Link (APML)**
  - 4.1 Overview
    - 4.1.1 Definitions
  - 4.2 SBI Bus Characteristics
    - 4.2.1 SMBus Protocol Support
    - 4.2.2 I2C Support
  - 4.3 SBI Processor Information
    - 4.3.1 SBI Processor Pins
      - 4.3.1.1 Physical Layer Characteristics
    - 4.3.2 Processor States
  - 4.4 SBI Protocols
    - 4.4.1 SBI Modified Block Write-Block Read Process Call
    - 4.4.2 SBI Remote Management Interface (SB-RMI)
      - 4.4.2.1 SB-RMI Processor State Access
        - 4.4.2.1.1 SB-RMI Read Processor Register and Read CPUID Commands
        - 4.4.2.1.2 SB-RMI Write Processor Register Command
        - 4.4.2.1.3 SB-RMI Protocol Status Codes
      - 4.4.2.2 SB-RMI Mailbox Service
        - 4.4.2.2.1 SB-RMI Mailbox Sequence
      - 4.4.2.3 SB-RMI Boot code status
      - 4.4.2.4 SB-RMI Register Access

- 4.4.2.4.1 SB-RMI Register Block Access
- 4.4.2.4.2 SB-RMI Register Byte Access
- 4.4.2.5 SB-RMI Alert
- 4.4.3 SBI Error Detection and Recovery
  - 4.4.3.1 Error Detection
    - 4.4.3.1.1 ACK/NAK Mechanism
    - 4.4.3.1.2 Packet Error Correction (PEC)
    - 4.4.3.1.3 Bus Timeouts
  - 4.4.3.2 Error Recovery
    - 4.4.3.2.1 SBI Bus Reset
- 4.5 SBI Physical Interface
  - 4.5.1 SBI SMBus Address
  - 4.5.2 SBI Bus Timing
- 4.6 SB-RMI Registers
- 5 SB Temperature Sensor Interface (SB-TSI)**
  - 5.1 Overview
    - 5.1.1 Definitions
  - 5.2 SB-TSI Protocol
    - 5.2.1 SB-TSI Send/Receive Byte Protocol
      - 5.2.1.1 SB-TSI Address Pointer
    - 5.2.2 SB-TSI Read/Write Byte Protocol
    - 5.2.3 Alert Behavior
    - 5.2.4 Atomic Read Mechanism
    - 5.2.5 SB-TSI Temperature and Threshold Encodings
    - 5.2.6 SB-TSI Temperature Offset Encoding
  - 5.3 SB-TSI Physical Interface
    - 5.3.1 SB-TSI SMBus Address
    - 5.3.2 SB-TSI Bus Timing
    - 5.3.3 SB-TSI Bus Electrical Parameters
    - 5.3.4 Pass-FET Option
  - 5.4 SB-TSI Registers
- 6 DXIO**
  - 6.1 DXIO Subsystem Overview
    - 6.1.1 Definitions
- 7 Fusion Controller Hub (FCH)**
  - 7.1 FCH Overview
    - 7.1.1 Acronyms
    - 7.1.2 Functional
    - 7.1.3 Memory Mapped IO (MMIO) Programming for Legacy Devices
      - 7.1.3.1 Description for FCH::IO::PCIInterrupt Map
    - 7.1.4 EMMC
    - 7.1.5 eSPI
    - 7.1.6 I3C Master
    - 7.1.7 GPIO
      - 7.1.7.1 Interrupt GPIO
      - 7.1.7.2 Wakeup GPIO
      - 7.1.7.3 Pure GPIO
    - 7.1.8 Sticky/non-sticky attribute of PMIO registers
    - 7.1.9 CLK IP Overview
      - 7.1.9.1 iCLK/eCLK mode
      - 7.1.9.2 CGPLL SSC Enable/Disable
        - 7.1.9.2.1 CGPLL -0.375% SSC Programming
        - 7.1.9.2.2 CGPLL -0.3% SSC Programming

- 7.1.9.3 Global A-Link/B-Link Clock Gating
- 7.1.9.4 GPP CLKREQB Mapping Table
- 7.1.10 FCH Register Access Information Guide
- 7.1.11 Reset Overview
- 7.1.12 RTC
  - 7.1.12.1 RTC Register
  - 7.1.12.2 External RTC Decode
- 7.1.13 Strap Definition
- 7.1.14 Address Mapping Table
- 7.2 FCH BTS Check Table
- 7.3 Registers
  - 7.3.1 Legacy Block Configuration Registers (IO)
    - 7.3.1.1 Registers
  - 7.3.2 I/O Advanced Programmable Interrupt Control
    - 7.3.2.1 IOAPIC Registers
  - 7.3.3 SMI Registers
  - 7.3.4 High Precision Event Timer (HPET) Registers
  - 7.3.5 Watchdog Timer (WDT) Registers
  - 7.3.6 Wake Alarm Device (AcDcTimer) Registers
  - 7.3.7 Always On Always Connected (AOAC) Registers
  - 7.3.8 ISA Bridge
    - 7.3.8.1 eSPI Registers
  - 7.3.9 MISC Registers
    - 7.3.9.1 Miscellaneous (MISC) Registers
    - 7.3.9.2 Power Management (PM) Registers and Standard ACPI Registers
    - 7.3.9.3 RTC External Registers
    - 7.3.9.4 Power Management (PM2) Registers
  - 7.3.10 GPIO Pin control registers
    - 7.3.10.1 IOMUX Registers
      - 7.3.10.1.1 IOMUX pin – EGPIO145/146/147/148 Programming Guideline
      - 7.3.10.1.2 IOMUX Funtional Table
    - 7.3.10.2 GPIO Registers

# List of Figures

Figure 1:	Adobe® Reader Hyperlink Opens New Window Configuration
Figure 2:	Adobe® Reader Select Between Opened Files
Figure 3:	Adobe® Reader Searching a Multiple Volume PPR
Figure 4:	Adobe® Reader Find Current Bookmark Button
Figure 5:	Register Physical Mnemonic, Title, and Name
Figure 6:	Full Width Register Attributes
Figure 7:	Register Description
Figure 8:	Register Instance Table: Content Ordering in a Row
Figure 9:	Register Instance Table: MSR Example
Figure 10:	Register Instance Table: MSR Range Example
Figure 11:	Register Instance Table: BAR as Register Reference
Figure 12:	Register Instance Table: Bus Implied to be 00h
Figure 13:	Register Instance Table: Data Port Select
Figure 14:	Simple Register Field Example
Figure 15:	Register Field Sub-Row for {Reset,AccessType,Init,Check}
Figure 16:	Register Field Sub-Row for Description
Figure 17:	Register Field Sub-Row for Valid Value Table
Figure 18:	Register Field Sub-Row for Valid Bit Table
Figure 19:	Revision History Format Example
Figure 20:	Change Notation Example
Figure 21:	Register Sharing Domains
Figure 22:	Instance Parameters
Figure 23:	SBI Transmission Protocol
Figure 24:	RTS Thermal Management Example
Figure 25:	SB-TSI Thermal Management Example
Figure 26:	Alert Assertion Diagram
Figure 27:	Pass FET Implementation

# List of Tables

Table 1:	Reference Documents Listing
Table 2:	Arithmetic and Logical Operator Definitions
Table 3:	Function Definitions
Table 4:	Operator Precedence and Associativity
Table 5:	Register Mnemonic Definitions
Table 6:	Logical Mnemonic Definitions
Table 7:	Physical Mnemonic Definitions
Table 8:	AccessType Definitions
Table 9:	Reset Type Definitions
Table 10:	Init Type Definitions
Table 11:	Definitions
Table 12:	Package Definitions
Table 13:	Definitions
Table 14:	MPB Definition
Table 15:	SMM Initial State
Table 16:	SMM Save State
Table 17:	ICR Valid Combinations
Table 18:	Guidance for Common Performance Statistics with Complex Event Selects
Table 19:	Guidance for Pipeline Utilization Analysis Statistics
Table 20:	PMC_Definitions
Table 21:	Machine Check Terms and Acronyms
Table 22:	Legacy MCA MSR Layout
Table 23:	MCAX MSR Layout
Table 24:	MCAX Implementation-Specific Register Layout
Table 25:	Error Overwrite Priorities
Table 26:	Error Scope Hierarchy
Table 27:	Error Code Types
Table 28:	Error code: transaction type (TT)
Table 29:	Error codes: cache level (LL)
Table 30:	Error codes: memory transaction type (RRRR)
Table 31:	Blocks Capable of Supporting MCA Banks
Table 32:	Mapping of Blocks to MCA_IPID[HwId] and MCA_IPID[McaType]
Table 33:	Legacy MCA Registers
Table 34:	MCAX Registers
Table 35:	Core MCA Bank to Block Mapping
Table 36:	Non-core MCA Bank to Block Mapping
Table 37:	MCA_STATUS_LS
Table 38:	MCA_ADDR_LS
Table 39:	MCA_SYND_LS
Table 40:	MCA_STATUS_IF
Table 41:	MCA_ADDR_IF
Table 42:	MCA_SYND_IF
Table 43:	MCA_STATUS_L2
Table 44:	MCA_ADDR_L2
Table 45:	MCA_SYND_L2
Table 46:	MCA_STATUS_DE
Table 47:	MCA_ADDR_DE
Table 48:	MCA_SYND_DE
Table 49:	MCA_STATUS_EX
Table 50:	MCA_ADDR_EX

Table 51:	MCA_SYND_EX
Table 52:	MCA_STATUS_FP
Table 53:	MCA_ADDR_FP
Table 54:	MCA_SYND_FP
Table 55:	MCA_STATUS_L3
Table 56:	MCA_ADDR_L3
Table 57:	MCA_SYND_L3
Table 58:	MCA_STATUS_CS
Table 59:	MCA_ADDR_CS
Table 60:	MCA_SYND_CS
Table 61:	MCA_STATUS_PIE
Table 62:	MCA_ADDR_PIE
Table 63:	MCA_SYND_PIE
Table 64:	MCA_STATUS_UMC
Table 65:	MCA_ADDR_UMC
Table 66:	MCA_SYND_UMC
Table 67:	MCA_STATUS_NBIO
Table 68:	MCA_ADDR_NBIO
Table 69:	MCA_SYND_NBIO
Table 70:	APML Definitions
Table 71:	SB-RMI Functions
Table 72:	SB-RMI Read Processor Register Command Protocol
Table 73:	SB-RMI Read CPUID Command Protocol
Table 74:	SB-RMI Read Data/Status Command Protocol
Table 75:	SB-RMI Load Address Command Protocol
Table 76:	SB-RMI Write Processor Register Command Protocol
Table 77:	SB-RMI Status Codes
Table 78:	SB-RMI Soft Mailbox Message
Table 79:	SB-RMI Soft Mailbox Error Code
Table 80:	SB-RMI Register Block Write Protocol
Table 81:	SB-RMI Register Block Read Protocol
Table 82:	SB-RMI Register Write Byte Protocol
Table 83:	SB-RMI Register Read Byte Protocol
Table 84:	SB-TSI Definitions
Table 85:	SB-TSI CPU Temperature and Threshold Encoding Examples
Table 86:	SB-TSI Temperature Offset Encoding Examples
Table 87:	SB-TSI Address Encodings
Table 88:	Link Definitions
Table 89:	List of Acronyms used in FCH
Table 90:	ValidValuesTable: PCI interrupt index list of PCI_INTR_INDEX bit[6:0]
Table 91:	I3C PAD Control Configuration
Table 92:	PMIO Register Sticky/Non-sticky Attributes
Table 93:	GPP ClkREQB Mapping
Table 94:	Register Access Information
Table 95:	Reset Type
Table 96:	Reset Overview
Table 97:	Strap Pins/Function
Table 98:	Address Space Mapping under APB BUS
Table 99:	FCH BTS Common Table
Table 100:	FCH Device D3 control/status mapping
Table 101:	ACPI MMIO Space Allocation
Table 102:	IOMUX Function Table
Table 103:	I2C Pad Configuration Method



Table 104:	GPIO BANK0 Register Default Value
Table 105:	GPIO BANK1 Register Default Value
Table 106:	GPIO BANK2 Register Default Value
Table 107:	GPIO BANK3 Register Default Value
Table 108:	Debounce Timer Definition

## 1 Overview

### 1.1 Intended Audience

This document provides the processor behavioral definition and associated design notes. It is intended for platform designers and for programmers involved in the development of BIOS functions, drivers, and operating system kernel modules.

### 1.2 Reference Documents

*Table 1: Reference Documents Listing*

Term	Description
<b>docAPM1</b>	AMD64 Architecture Programmer's Manual Volume 1: Application Programming, order# 24592.
<b>docAPM2</b>	AMD64 Architecture Programmer's Manual Volume 2: System Programming, order# 24593.
<b>docAPM3</b>	AMD64 Architecture Programmer's Manual Volume 3: Instruction-Set Reference, order# 24594.
<b>docAPM4</b>	AMD64 Architecture Programmer's Manual Volume 4: 128-Bit and 256-Bit Media Instructions, order# 26568.
<b>docAPM5</b>	AMD64 Architecture Programmer's Manual Volume 5: 64-Bit Media and x87 Floating-Point Instructions, order# 26569.
<b>docACPI</b>	Advanced Configuration and Power Interface (ACPI) Specification. <a href="http://uefi.org/specifications">http://uefi.org/specifications</a> .
<b>docASF</b>	Alert Standard Format Specification. <a href="http://dmtf.org/standards/asf">http://dmtf.org/standards/asf</a> .
<b>docDP</b>	VESA DisplayPort Standard. <a href="http://www.vesa.org/vesa-standards">http://www.vesa.org/vesa-standards</a> .
<b>docIOMMU</b>	AMD I/O Virtualization Technology Specification, order# 48882.
<b>docI2C</b>	I2C Bus Specification. <a href="http://www.nxp.com/documents/user_manual/UM10204.pdf">http://www.nxp.com/documents/user_manual/UM10204.pdf</a>
<b>docJEDEC</b>	JEDEC Standards. <a href="http://www.jedec.org">http://www.jedec.org</a> .
<b>docPCIe</b>	PCI Express® Specification. <a href="http://www.pcisig.org">http://www.pcisig.org</a> .
<b>docPCIb</b>	PCI Local Bus Specification. <a href="http://www.pcisig.org">http://www.pcisig.org</a> .
<b>docSDHC</b>	Secure Digital Host Controller Standard Specification. <a href="https://www.sdcard.org">https://www.sdcard.org</a> .
<b>docUSB</b>	Universal Serial Bus Specification. <a href="http://www.usb.org">http://www.usb.org</a> .

#### 1.2.1 Documentation Conventions

When referencing information found in external documents listed in Reference Documents, the "=>" operator is used. This notation represents the item to be searched for in the reference document. For example:

docExDoc => Header1 => Header2

is to have the reader use the search facility when opening referenced document "docExDoc" and search for "Header2". "Header2" may appear more than once in "docExDoc", therefore, referencing the one that follows "Header1". In that case, the easiest way to get to Header2 is to use the search to locate Header1, then again to locate "Header2".

### 1.3 Adobe® Reader

This section describes how to configure and use Adobe® Reader for the PPR PDFs.

Adobe Reader is the recommended tool for viewing PPR pdfs and can be downloaded at <https://get.adobe.com/reader/>.

### 1.3.1 Adobe® Reader Configuration

This section describes how to configure Adobe Reader for the PPR PDFs.

#### 1.3.1.1 Open Hyperlink Document in New Window

The Open Hyperlink Document in New Window setting opens a new window for a hyperlink, instead of opening the hyperlink document in the same window.

Menu->Preferences:

- Documents
  - Open Settings:
    - Deselect: Open cross-document links in same window

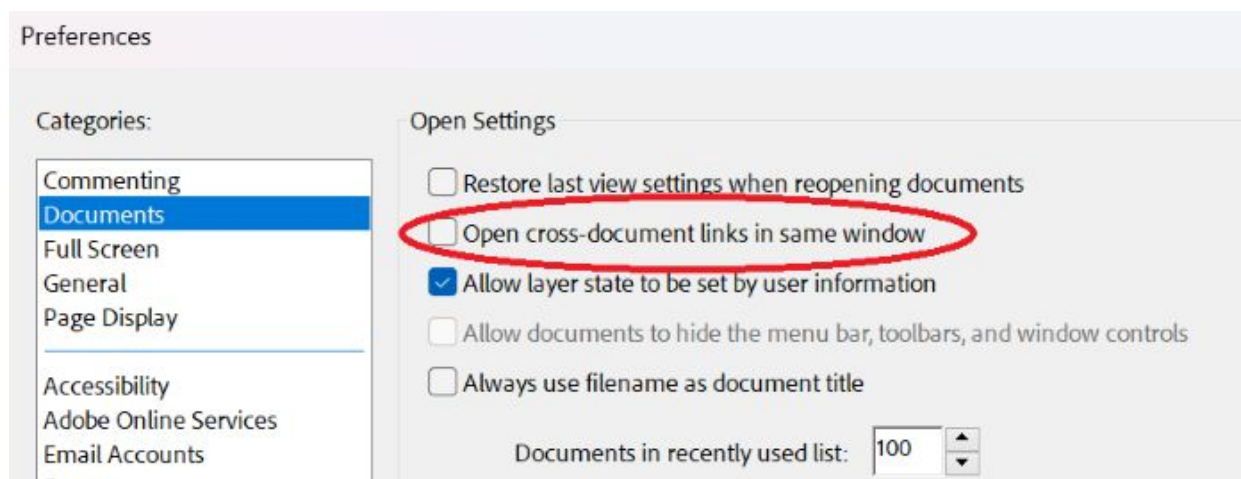


Figure 1: Adobe® Reader Hyperlink Opens New Window Configuration

The following Figure shows how when hyperlinking from one document to another, the original document is left open. The tab that is not grayed-out indicates the foreground window.

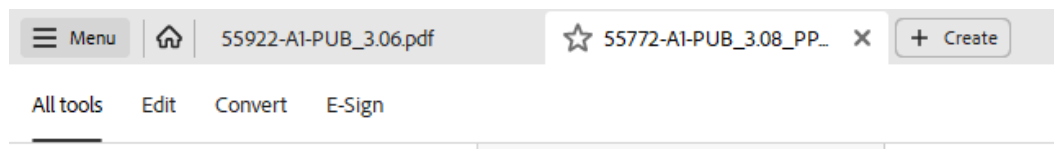


Figure 2: Adobe® Reader Select Between Opened Files

### 1.3.2 Adobe® Reader Usage

This section describes how to use Adobe Reader for the PPR PDFs.

NOTE: PDF's are distributed in zip format. In order to search and hyperlink between PDF volumes, the zip contents must be extracted to a folder.

### 1.3.2.1 Searching a Multiple Volume PPR

The PPR is a multiple PDF document and searching all PDFs is performed as follows:

- The zip of PDF files must be extracted to a directory where the search will be performed. A search across multiple PDF files can not be performed from within a zip of PDF's.
- Open search by selecting Menu->Search->Advanced Search (Shift+Ctrl+F)
- Select "All PDF Documents in" and select "Browse for Location...", which opens the "Browse For Folder" window.
- In the "Browse For Folder" window, select the folder that contains the PPR PDFs that need to be searched, and select OK.

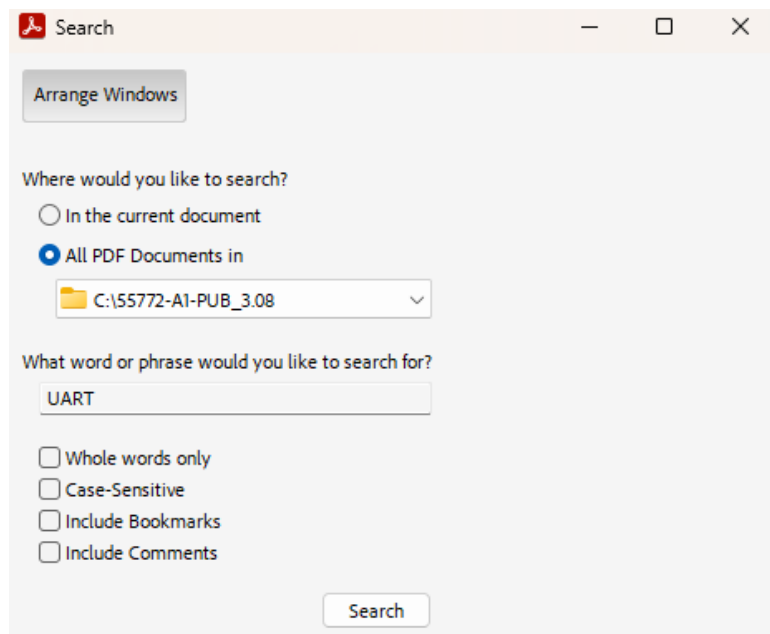


Figure 3: Adobe® Reader Searching a Multiple Volume PPR

### 1.3.2.2 Cross-References and Hyperlinks

A cross-reference is a link to a location within the same PDF. A hyperlink is a link to a location within a different PDF.

- For cross-references, use "Previous View" to return from the current location to the previous location.
  - Menu->View->Page Navigation->Previous View
- Hyperlinks between documents leave the current location unchanged in the PDF that contained the hyperlink.
- In order for hyperlinks to work properly the zip of PDF's must be extracted to a directory. Hyperlinks will not function within a zip of PDF's.

### 1.3.2.3 Find Current Bookmark

The bookmark pane can highlight the current bookmark associated with the viewer pane by selecting the "find current bookmark" button, as shown below.

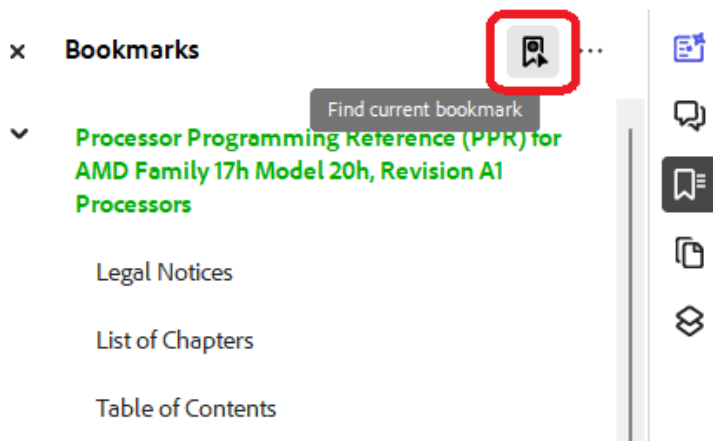


Figure 4: Adobe® Reader Find Current Bookmark Button

## 1.4 Conventions

### 1.4.1 Numbering

- Binary numbers: Binary numbers are indicated either by appending a "b" at the end (e.g., 0110b) or by Verilog syntax (e.g., 4'b0110).
- Hexadecimal numbers: Hexadecimal numbers are indicated by appending an "h" to the end (e.g., 45F8h) or by Verilog syntax (e.g., 16'h45F8).
- Decimal numbers: A number is decimal if not specified to be binary or hex.
- Exception: Physical register mnemonics are implied to be hex without the h suffix.
- Underscores in numbers: Underscores are used to break up numbers to make them more readable. They do not imply any operation (e.g., 0110\_1100).

### 1.4.2 Arithmetic And Logical Operators

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

Table 2: Arithmetic and Logical Operator Definitions

Operator	Definition
{}	Concatenation. Curly brackets are used to indicate a group of bits that are concatenated together. Each set of bits is separated by a comma (e.g., {Addr[3:2], Xlate[3:0]} represents a 6-bit values; the two MSBs are Addr[3:2] and the four LSBs are Xlate[3:0]).
	Bitwise OR (e.g., 01b   10b == 11b).
	Logical OR (e.g., 01b    10b == 1b). It treats a multi-bit operand as 1 if >= 1 and produces a 1-bit result.
&	Bitwise AND (e.g., 01b & 10b == 00b).
&&	Logical AND (e.g., 01b && 10b == 1b). It treats a multi-bit operand as 1 if >= 1 and produces a 1-bit result.
^	Bitwise exclusive-OR (e.g., 01b ^ 10b == 11b). Sometimes used as "raised to the power of" as well, as indicated by the context in which it is used (e.g., 2^2 == 4).
~	Bitwise NOT (also known as one's complement). (e.g., ~10b == 01b).
!	Logical NOT (e.g., !10b == 0b). It treats a multi-bit operand as 1 if >= 1 and produces a 1-bit result.

<, <=, >, >=, ==, !=	Relational. Less than, Less than or equal, greater, greater than or equal, equal, and not equal.
+, -, *, /, %	Arithmetic. Addition, subtraction, multiplication, division, and modulus.
<<	Bitwise left shift. Shift left first operand by the number of bits specified by the 2nd operand (e.g., 01b << 01b == 10b).
>>	Bitwise right shift. Shift right first operand by the number of bits specified by the 2nd operand (e.g., 10b >> 01b == 01b).
?:	Ternary conditional (e.g., condition ? value if true : value if false).

Table 3: Function Definitions

Term	Description
<b>ABS</b>	ABS(integer expression): Remove sign from signed value.
<b>FLOOR</b>	FLOOR(integer expression): Rounds real number down to nearest integer.
<b>CEIL</b>	CEIL(real expression): Rounds real number up to nearest integer.
<b>MIN</b>	MIN(integer expression list): Picks minimum integer or real value of comma separated list.
<b>MAX</b>	MAX(integer expression list): Picks maximum integer or real value of comma separated list.
<b>COUNT</b>	COUNT(integer expression): Returns the number of binary 1's in the integer.
<b>ROUND</b>	ROUND(real expression): Rounds to the nearest integer; halfway rounds away from zero.
<b>UNIT</b>	UNIT(register field reference): Input operand is a register field reference that contains a valid values table that defines a value with a unit (e.g., clocks, ns, ms, etc.). This function takes the value in the register field and returns the value associated with the unit (e.g., If the field had a valid value definition where 1010b was defined as 5 ns). Then if the field had the value of 1010b, then UNIT() would return the value 5.
<b>POW</b>	POW(base, exponent): POW(x,y) returns the value x to the power of y.

#### 1.4.2.1 Operator Precedence and Associativity

This document follows C operator precedence and associativity. The following table lists operator precedence (highest to lowest). Their associativity indicates in what order operators of equal precedence in an expression are applied. Parentheses are also used to group subexpressions to force a different precedence; such parenthetical expressions can be nested and are evaluated from inner to outer (e.g., "X = A || !B && C" is the same as "X = A || ((!B) && C)").

Table 4: Operator Precedence and Associativity

Operator	Description	Associativity
!, ~	Logical negation/bitwise complement	right to left
*, /, %	Multiplication/division/modulus	left to right
+, -	Addition/subtraction	left to right
<<, >>	Bitwise shift left, Bitwise shift right	left to right
<, <=, >, >=, ==, !=	Relational operators	left to right
&	Bitwise AND	left to right
^	Bitwise exclusive OR	left to right
	Bitwise inclusive OR	left to right
&&	Logical AND	left to right
	Logical OR	left to right
?:	Ternary conditional	right to left

### 1.4.3 Register Mnemonics

A register mnemonic is a short name that uniquely refers to a register, either all instances of that register, some instances, or a single instance.

Every register instance can be expressed in 2 forms, logical and physical, as defined below.

*Table 5: Register Mnemonic Definitions*

Term	Description
<b>logical mnemonic</b>	The register mnemonic format that describes the register functionally, what namespace to which the register belongs, a name for the register that connotes its function, and optionally, named parameters that indicate the different function of each instance (e.g., Link::Phy::PciDevVendIDF3). See 1.4.3.1 [Logical Mnemonic].
<b>physical mnemonic</b>	The register mnemonic that is formed based on the physical address used to access the register (e.g., D18F3x00). See 1.4.3.2 [Physical Mnemonic].

#### 1.4.3.1 Logical Mnemonic

The logical mnemonic format consists of a register namespace, a register name, and optionally a register instance specifier (e.g., register namespace::register name register instance specifier).

For Unb::PciDevVendIDF3:

- The register namespace is Unb, which is the UNB IP register namespace.
- The register name is PciDevVendIDF3, which reads as PCICFG device and vendor ID in Function 3.
- There is no register instance specifier because there is just a single instance of this register.

For Dct::Phy::CalMisc2\_dct[1:0]\_chiplet[BCST,3:0]\_pad[BCST,11:0]:

- The register namespace is Dct::Phy, which is the DCT PHY register namespace.
- The register name is CalMisc2, which reads as miscellaneous calibration register 2.
- The register instance specifier is \_dct[1:0]\_chiplet[BCST,3:0]\_pad[BCST,11:0], which indicates that there are 2 DCTPHY instances, each IP for this register has 5 chiplets (0-3 and BCST), and for each chiplet 13 pads (0-11 and BCST). This register has 130 instances. (2\*5\*13)

Table 6: Logical Mnemonic Definitions

Term	Description
<b>register namespace</b>	A namespace for which the register name must be unique. A register namespace indicates to which IP it belongs and an IP may have multiple namespaces. A namespace is a string that supports a list of "::" separated names. The convention is for the list of names to be hierarchical, with the most significant name first and the least significant name last (e.g., Link::Phy::Rx is the RX component in the Link PHY).
<b>register name</b>	A name that connotes the function of the register.
<b>register instance specifier</b>	The register instance specifier exists when there is more than one instance for a register. The register instance specifier consists of one or more register instance parameter specifier (e.g., The register instance specifier <code>_dct[1:0]_chiplet[BCST,3:0]_pad[BCST,11:0]</code> consists of 3 register instance parameter specifiers, <code>_dct[1:0]</code> , <code>_chiplet[BCST,3:0]</code> , and <code>_pad[BCST,11:0]</code> ).
<b>register instance parameter specifier</b>	A register instance parameter specifier is of the form <code>_register parameter name[register parameter value list]</code> (e.g., The register instance parameter specifier <code>_dct[1:0]</code> has a register parameter name of <code>dct</code> (The DCT PHY instance name) and a register parameter value list of "1:0" or 2 instances of DCT PHY).
<b>register parameter name</b>	A register parameter name is the name of the number of instances at some level of the logical hierarchy (e.g., The register parameter name <code>dct</code> specifies how many instances of the DCT PHY exist).
<b>register parameter value list</b>	The register parameter value list is the logical name for each instance of the register parameter name (e.g., For <code>_dct[1:0]</code> , there are 2 DCT PHY instances, with the logical names 0 and 1, but it should be noted that the logical names 0 and 1 can correspond to physical values other than 0 and 1). It is the purpose of the <code>AddressMappingTable</code> to map these register parameter values to physical address values for the register.

#### 1.4.3.2 Physical Mnemonic

The physical register mnemonic format varies by the access method. The following table describes the supported physical register mnemonic formats.



Table 7: Physical Mnemonic Definitions

Term	Description
<b>PCICFG</b>	The PCICFG, or PCI defined configuration space, physical register mnemonic format is of the form DXFYxZZZ. Bus 0 is implied, X specifies the hexadecimal device number (this may be 1 or 2 digits). Y specifies the function number. ZZZ specifies the hexadecimal byte address (this may be 2 or 3 digits). Example: D18F2x40 specifies the register at bus 0, device 18h, function 2, and address 40h. If the mnemonic starts with B, then the physical mnemonic format is BWDXFYxZZZ where WW specifies the hexadecimal bus number (1 or 2 hex digits) or "XX" implying that the bus is relocatable. Example; BXXD00F6x40 specifies that the bus is relocatable, B0AD00F2x000 specifies that the bus is 0Ah.
<b>BAR</b>	The BAR, or base address register, physical register mnemonic format is of the form PREFIXxZZZ. PREFIX is an all capital letter name that connotes the BAR to which the offset is added to get the physical address of the operation. ZZZ is the offset.
<b>MSR</b>	The MSR, or x86 model specific register, physical register mnemonic format is of the form MSRXXXX_XXXX, where XXXX_XXXX is the hexadecimal MSR number. This space is accessed through x86 defined RDMSR and WRMSR instructions.
<b>PMC</b>	The PMC, or x86 performance monitor counter, physical register mnemonic format is any of the forms {PMCxXXX, L2IPMCxXXX, NBPMCxXXX}, where XXX is the performance monitor select.
<b>CPUID</b>	The CPUID, or x86 processor identification state, physical register mnemonic format is of the form CPUID FnXXXX_XXXX_EiX[_xYYY], where XXXX_XXXX is the hex value in the EAX and YYY is the hex value in ECX.

#### 1.4.4 Register Format

A register is a group of register instances that have the same field format (same bit indices and field names).

##### 1.4.4.1 Register Instances

All instances of a register:

- Have the same:
  - Field bit indices and names
  - Field titles, descriptions, valid values.
  - Register title
  - Register description
- Fields may have different: (instance specific)
  - Access Type. See 1.4.4.10 [Field Access Type].
  - Reset. See 1.4.4.11 [Field Reset].
  - Init. See 1.4.4.12 [Field Initialization].
  - Check. See 1.4.4.13 [Field Check].

##### 1.4.4.2 Register Physical Mnemonic, Title, and Name

A register definition is identified by a table that starts with a heavy bold line. The information above the bold line in order is:

1. The physical mnemonic of the register.
  - A register that has multiple instances, may have instances that have different access methods, each with it's own physical mnemonic format.
  - In the event that there are multiple physical mnemonic formats, the physical mnemonic format chosen is the most commonly used physical mnemonic.

- The physical mnemonic is not intended to represent the physical mnemonics of all instances of the register. It is only a visual aid to identify a register when scanning down a list, for readers that prefer to find registers by physical mnemonic. If "..." occurs in the physical mnemonic, the range is first ... last. There is no implication as to how many instances exist between first and last. See 1.4.4.5 [Register Instance Table].
- 2. The register title in brackets.
- 3. The register name in parenthesis.

Physical Mnemonic	Title	Name
MSR0000_0010	[Time Stamp Counter]	(TSC)
Read-write, Volatile. Reset: 0000_0000_0000_0000h.		
Core::X86::Msr::TSC_three[1:0]_core[3:0]_thread[1:0]; MSR00000010		
Bits	Description	
63:0	TSC: <b>time stamp counter</b> . Read-write, Volatile. Reset: 0. The TSC increments at the P0 frequency. The TSC counts at the same rate in all P-states, all C states, S0, or S1. A read of this MSR in guest mode is affected by Core::X86::Msr::TscRateMsr. The value (TSC/TSCRatio) is the TSC P0 frequency based value (as if TSCRatio == 1.0) when (TSCRatio != 1.0).	

Figure 5: Register Physical Mnemonic, Title, and Name

#### 1.4.4.3 Full Width Register Attributes

The first line that follows the bold line contains the attributes that apply to all fields of the register. This row is rendered as a convenience to the reader and replicates content that exists in the register field.

- AccessType: If all non-reserved fields of a register have the same access type, then the access type is rendered in this row.
  - The supported access types are specified by 1.4.4.10 [Field Access Type].
  - The example figure shows that the access type "Read-write, Volatile" applies to all non-reserved fields of the register.
- Reset: If all non-reserved fields of a register have a constant reset and are all the same type (Warm, Cold, Fixed), then the full width register reset is rendered in this row. The example figure shows the reset "0000\_0000\_0000\_0000h". See 1.4.4.11 [Field Reset].
  - The value zero (0) is assumed for display purposes for all reserved fields.
- If none of the above content is rendered, then this row of the register is not rendered.

#### MSR0000\_0010 [Time Stamp Counter] (TSC)

Read-write, Volatile. Reset: 0000_0000_0000_0000h.	
Core::X86::Msr::TSC_three[1:0]_core[3:0]_thread[1:0]; MSR00000010	
Bits	Description
63:0	TSC: <b>time stamp counter</b> . Read-write, Volatile. Reset: 0. The TSC increments at the P0 frequency. The TSC counts at the same rate in all P-states, all C states, S0, or S1. A read of this MSR in guest mode is affected by Core::X86::Msr::TscRateMsr. The value (TSC/TSCRatio) is the TSC P0 frequency based value (as if TSCRatio == 1.0) when (TSCRatio != 1.0).

Figure 6: Full Width Register Attributes

#### 1.4.4.4 Register Description

The register description is optional and appears after the "full width register attributes" row and before the "register

instance table" rows. The register description can be one or more paragraphs.

#### PciDevVendIDF3 [Device/Vendor ID]

Read-only. Reset: 0000_1022h.	
A register description. That can be multiple paragraphs.	
Link::Phy::Tx::PciDevVendIDF3; D18F3x00	
Bits	Description
31:16	DeviceID: device ID. Read-only. Reset: Fixed,0000h.
15:0	VendorID: vendor ID. Read-only. Reset: Fixed,1022h. Init: 1234h.

Figure 7: Register Description

### 1.4.4.5 Register Instance Table

The zero or more rows of 8-pt font before the Bits/Description row is the register instance table.

The register instance table can generally be described as follows:

- Each row describes the access method of one or more register instances.
- If a row describes two or more instances, then the logical instance range, left to right, corresponds to the physical range, left to right.
- The absence of register instance rows indicates that the register exists for documentation purposes, and no access method is described for the register.

Because there are multiple access methods for all the registers, each of the following subsections describes an aspect of the register instance table in isolation.

#### 1.4.4.5.1 Content Ordering in a Row

Content in a register instance table row is ordered as follows:

- The text up to the first semicolon is the logical mnemonic.
  - See 1.4.3.1 [Logical Mnemonic].
- The text after the first semicolon is the physical mnemonic.
  - See 1.4.3.2 [Physical Mnemonic].
- Optionally, content after the physical mnemonic provides additional information about the access method for the register instances in the row.

#### BXXD00F0x000 (NB\_VENDOR\_ID)

Read-only. Reset: 1022h.	
Vendor ID Register	
IOHC::NB_VENDOR_ID_aliasHOST; BXXD00F0x000; BXX=IOHC::NB_BUS_NUM_CNTL_aliasSMN[NB_BUS_NUM]	
IOHC::NB_VENDOR_ID_aliasSMN; NBCFGx00000000; NBCFG=13B0_0000h	

Figure 8: Register Instance Table: Content Ordering in a Row

### 1.4.4.5.2 Multiple Instances Per Row

Multiple instances in a row is represented by a single dimension "range" in the logical mnemonic and the physical mnemonic.

The single dimension order of instances is the same for both the logical and physical mnemonic. The first logical mnemonic is associated with the first physical mnemonic, so forth for the 2nd, up until the last.

- Brackets indicates a list, most significant to least significant.
- The ":" character indicates a continuous range between 2 values.
- The "," character separates non-contiguous values.
- There are some cases where more than one logical mnemonic maps to a single physical mnemonic.

Note that it is implied that the MSR {lthree,core,thread} parameters are not part of a range.

Example:

NAMESP::REGNAME\_inst[BLOCK[5:0],BCST]\_aliasHOST; FFF1x00000088\_x[000[B:6]\_0001,00000000]

- There are 7 instances.
- NAMESP is the namespace.
- 6 instances are represented by the sub-range 000[B:6]\_0001.
- \_instBCST corresponds to FFF1x00000088\_x00000000.
- \_inst BLOCK 0 corresponds to FFF1x00000088\_x00060001.
- ...
- \_inst BLOCK 5 corresponds to FFF1x00000088\_x000B0001.

#### 1.4.4.5.3 MSR Access Method

The MSR parameters {lthree,core,thread} are implied by the identity of the core on which the RDMSR/WRMSR is being executed, and therefore are not represented in the physical mnemonic.

MSRs that are:

- per-thread have the {lthree,core,thread} parameters.
- per-core do not have the thread parameter.
- per-L3 do not have the {core,thread} parameters.
- common to all L3's do not have the {lthree,core,thread} parameters.

##### 1.4.4.5.3.1 MSR Per-Thread Example

An MSR that is per-thread has all three {lthree,core,thread} parameters and all instances have the same physical mnemonic.

#### MSR0000\_0010 [Time Stamp Counter] (TSC)

Read-write, Volatile. Reset: 0000_0000_0000_0000h.	
Core::X86::Msr::TSC_lthree[1:0]_core[3:0]_thread[1:0]; MSR00000010	
Bits	Description
63:0	<b>TSC: time stamp counter.</b> Read-write, Volatile. Reset: 0. The TSC increments at the P0 frequency. The TSC counts at the same rate in all P-states, all C states, S0, or S1. A read of this MSR in guest mode is affected by Core::X86::Msr::TscRateMsr. The value (TSC/TSCRatio) is the TSC P0 frequency based value (as if TSCRatio == 1.0) when (TSCRatio != 1.0).

Figure 9: Register Instance Table: MSR Example

##### 1.4.4.5.3.2 MSR Range Example

An MSR can exist as a range for a parameter other than the {lthree,core,thread} parameters.

In the following example the n parameter is a range. The \_n0 value corresponds to MSR0000\_0201, and so on.

**MSR0000\_0201 [Variable-Size MTRRs Mask] (MtrrVarMask)**

Reset: 0000_0000_0000_0000h.
Core::X86::Msr::MtrrVarMask [n[7:0]] lthree[1:0]_core[3:0]; MSR0000_0201[[F,D,B,9,7,5,3,1]]

Figure 10: Register Instance Table: MSR Range Example

**1.4.4.5.4 BAR Access Method**

The BAR access method is indicated by a physical mnemonic that has the form PREFIXxNUMBER.

- BAR, which stands for "Base Address Register", is the base address for the IP block and can be a constant, a register field, or an expression that consists of one or more register fields.
- Example: APICx0000. The BAR prefix is "APIC".

The BAR prefix represents either a constant or an expression that consists of a register reference.

**1.4.4.5.4.1 BAR as a Register Reference**

A relocatable BAR is when the base of an IP is not a constant.

- The prefix NTBPRIBAR0 represents the base of the IP, the value of which comes from the register NBIFEPFNCFG::BASE\_ADDR\_1\_aliasHOST\_instNBIF0\_func1[BASE\_ADDR].
- The address of the register is NBIFEPFNCFG::BASE\_ADDR\_1\_aliasHOST\_instNBIF0\_func1[BASE\_ADDR] + 00000h.

**NTBPRIBAR0x00000 (NTB\_SMU\_PCTRL0)**

Reset: 0000_0000h.
NTB::NTB_SMU_PCTRL0_aliasHOSTPRI; NTBPRIBAR0x00000;
NTBPRIBAR0=NBIFEPFNCFG::BASE_ADDR_1_aliasHOST_instNBIF0_func1[BASE_ADDR]
NTB::NTB_SMU_PCTRL0_aliasHOSTSEC; NTBSECBAR0x00000;
NTBSECBAR0=NBIFEPFNCFG::BASE_ADDR_1_aliasHOST_instNBIF2_func1[BASE_ADDR]
NTB::NTB_SMU_PCTRL0_aliasSMN; NTBx00000000; NTB=0400_0000h

Figure 11: Register Instance Table: BAR as Register Reference

**1.4.4.5.5 PCICFG Access Method**

The PCICFG access method is indicated by a physical mnemonic that has the form DXXFXxNUMBER. There are 2 cases:

- Bus omitted and implied to be 00h.
- Bus represented as BXX and indicates that the bus is indicated by a register field.

Example:

- Example: D18F0x000. (The bus, when omitted, is implied to be 00h)
- Example: BXXD0F0x000. (The bus as an expression that includes a register reference)

**1.4.4.5.5.1 PCICFG Bus Implied to be 00h**

Example:

- The absence of a B before the D14 implies that the bus is 0.

FCH::ITF::LPC::PciDevVendID_aliasHOST; D14F3x000
--

Figure 12: Register Instance Table: Bus Implied to be 00h

#### 1.4.4.5.6 Data Port Access Method

A data port requires that the data port select be written before the register is accessed via the data port.

- The registers behind a data port is also called an indirect address space.
- The implied access method is to first write the data port select and then to read/write the register at the data port address.
- There are some cases where there are 2 or more data port selects.

Example:

- The data port select value follows the "\_x".
- The data port select register follows the "DataPortWrite=".
- The access method for \_instPIE0\_aliasHOST is:
  1. Write 0005\_0001h to DF::FabricConfigAccessControl.
  2. Read/Write the PCICFG address D18F0x040.

DF::FabricBlockInstanceCount_inst[PIE0,BCST]_aliasHOST; D18F0x040_x[00050001,00000000]; DataPortWrite=DF::FabricConfigAccessControl
DF::FabricBlockInstanceCount_inst[PIE0,BCST]_aliasSMN; DFF0x00000040_x[00050001,00000000]; DFF0=0001_C000h; DataPortWrite=DF::FabricConfigAccessControl

Figure 13: Register Instance Table: Data Port Select

#### 1.4.4.6 Register Field Format

The register field definition are all rows that follow the Bits/Description row. Each field row represents the definition of a bit range, with the bit ranges ordered from most to least significant. There are 2 columns, with the left column defining the field bit range, and the right column containing the field definition.

There are 2 field definition formats, simple and complex. If the description can be described in the simple one paragraph format then the simple format is used, else the complex format is used.

#### 1.4.4.7 Simple Register Field Format

The simple register format compresses all content into a single paragraph with the following implied order:

1. Field Name (required)
  - Allowed to be Reserved. See 1.4.4.9 [Field Name is Reserved].
  - "FFXSE" in the example figure.
2. Field Title
  - "fast FXSAVE/FRSTOR enable" in the example figure.
3. Field Access Type. See 1.4.4.10 [Field Access Type].
  - In the example figure the access type is "Read-write".
4. Field Reset. See 1.4.4.11 [Field Reset].
  - In the example figure the reset is warm reset and "0".
5. Field Init. See 1.4.4.12 [Field Initialization].
6. Field Check. See 1.4.4.13 [Field Check].
7. Field Valid Values. If the valid values are single bit (e.g., 0=, 1=). See 1.4.4.14 [Field Valid Values].
  - In the example figure the 1= definition begins with "Enables" and ends with "mechanism".

- In the example figure there is no 0= definition.
8. Field Description. If it is a single paragraph.
- In the example figure the field description begins with "This is" and ends with "afterwards".

All fields that do not exist are omitted.

14	<b>FFXSR: fast FXSAVE/FRSTOR enable.</b> Read-write. Reset: 0. 1=Enables the fast FXSAVE/FRSTOR mechanism. A 64-bit operating system may enable the fast FXSAVE/FRSTOR mechanism if (Core::X86::CpuId::FeatureExtIdEdx[FFXSR] == 1). This bit is set once by the operating system and its value is not changed afterwards.
----	--

Figure 14: Simple Register Field Example

#### 1.4.4.8 Complex Register Field Format

Content that cannot be expressed in the single paragraph format is broken out to a separate sub-row (a definition column row).

Additional sub-rows are added in the following order:

1. Complex expression for {Reset,AccessType,Init,Check}.
2. Instance specific {Reset,AccessType,Init,Check} values.
3. Description, if more than 1 paragraph.
4. Valid values, if more than 0=/1=. Or a Valid bit table. (see figure)

The following figure highlights a complex access type specification.

63:0	<b>APerfReadOnly: read-only actual core clocks counter.</b> Reset: 0. This register increments in proportion to the actual number of core clocks cycles while the core is in C0. See Core::X86::Msr::MPerfReadOnly. This register is not affected by writes to Core::X86::Msr::APERF.
	AccessType: Core::X86::Msr::HWCR[EffFreqReadOnlyLock] ? Read-only, Volatile : Read-write, Volatile.

Figure 15: Register Field Sub-Row for {Reset,AccessType,Init,Check}

The following figure highlights a complex description specification.

4	<b>INVDWBINVD: INVD to WBINVD conversion.</b> Read-write. Reset: 1. Check: 1. 1=Convert INVD to WBINVD.
	<b>Description:</b> This bit is required to be set for normal operation when any of the following are true: <ul style="list-style-type: none"> <li>• An L2 is shared by multiple threads.</li> <li>• An L3 is shared by multiple cores.</li> <li>• CC6 is enabled.</li> <li>• Probe filter is enabled.</li> </ul>

Figure 16: Register Field Sub-Row for Description

The following figure highlights a complex valid value table, used either when the field is more than 1 bit or when the definition is more than a single sentence.



2:1	<b>CpuWdtTimeBase:</b> CPU watchdog timer time base. Read-write. Reset: 0. Specifies the time base for the timeout period specified in CpuWdtCountSel.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00b</td><td>1.31ms</td></tr> <tr> <td>01b</td><td>1.28us</td></tr> <tr> <td>10b</td><td>Reserved (5ns)</td></tr> <tr> <td>11b</td><td>Reserved</td></tr> </table>	Value	Description	00b	1.31ms	01b	1.28us	10b	Reserved (5ns)	11b	Reserved
Value	Description										
00b	1.31ms										
01b	1.28us										
10b	Reserved (5ns)										
11b	Reserved										

Figure 17: Register Field Sub-Row for Valid Value Table

The following figure highlights a valid bit table which is used when each bit has a specific function.

55:52	Reserved.										
51:48	<b>SliceMask.</b> Read-write. Reset: 0.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Bit</th><th>Description</th></tr> <tr> <td>[0]</td><td>L3 Slice 0 mask.</td></tr> <tr> <td>[1]</td><td>L3 Slice 1 mask.</td></tr> <tr> <td>[2]</td><td>L3 Slice 2 mask.</td></tr> <tr> <td>[3]</td><td>L3 Slice 3 mask.</td></tr> </table>	Bit	Description	[0]	L3 Slice 0 mask.	[1]	L3 Slice 1 mask.	[2]	L3 Slice 2 mask.	[3]	L3 Slice 3 mask.
Bit	Description										
[0]	L3 Slice 0 mask.										
[1]	L3 Slice 1 mask.										
[2]	L3 Slice 2 mask.										
[3]	L3 Slice 3 mask.										

Figure 18: Register Field Sub-Row for Valid Bit Table

#### 1.4.4.9 Field Name is Reserved

When a register field name is Reserved, and it does not explicitly specify an access type, then the implied access type is "write-as-read".

- Reads must not depend on the read value.
- Writes must only write the value that was read.

#### 1.4.4.10 Field Access Type

The AccessType keyword is optional and specifies the access type for a register field. The access type for a field is a comma separated list of the following access types.



Table 8: AccessType Definitions

Term	Description
<b>Read-only</b>	Readable; writes are ignored.
<b>Read-write</b>	Readable and writable.
<b>Read</b>	Readable; must be associated with one of the following {Write-once, Write-1-only, Write-1-to-clear, Error-on-write}.
<b>Write-once</b>	Capable of being written once; all subsequent writes have no effect. If not associated with Read, then reads are undefined.
<b>Write-only</b>	Writable. Reads are undefined.
<b>Write-1-only</b>	Writing a 1 sets to a 1; Writing a 0 has no effect. If not associated with Read, then reads are undefined.
<b>Write-1-to-clear</b>	Writing a 1 clears to a 0; Writing a 0 has no effect. If not associated with Read, then reads are undefined.
<b>Write-0-only</b>	Writing a 0 clears to a 0; Writing a 1 has no effect. If not associated with Read, then reads are undefined.
<b>Error-on-read</b>	Error occurs on read.
<b>Error-on-write</b>	Error occurs on write.
<b>Error-on-write-0</b>	Error occurs on bitwise write of 0.
<b>Error-on-write-1</b>	Error occurs on bitwise write of 1.
<b>Inaccessible</b>	Not readable or writable (e.g., Hide ? Inaccessible : Read-Write).
<b>Configurable</b>	Indicates that the access type is configurable as described by the documentation.
<b>Unpredictable</b>	The behavior of both reads and writes is unpredictable.
<b>Reserved-write-as-1</b>	Reads are undefined. Must always write 1.
<b>Reserved-write-as-0</b>	Reads are undefined. Must always write 0.
<b>Volatile</b>	Indicates that a register field value may be modified by hardware, firmware, or microcode when fetching the first instruction and/or might have read or write side effects. No read may depend on the results of a previous read and no write may be omitted based on the value of a previous read or write. Not volatile indicates that software may service a read from the results of a previous read and that a write may be dropped if it's value matches the value previously read or written.

#### 1.4.4.10.1 Conditional Access Type Expression

The ternary operator can be used to express an access type that is conditional on an expression that can contain any of the following:

- A register field value
- A constant
- A definition

#### 1.4.4.11 Field Reset

The Reset keyword is optional and specifies the value for a register field at the time that hardware exits reset, before firmware initialization initiates.

Unless preceded by one of the following prefixes, the reset value is called warm reset and the value is applied at both warm and cold reset.

*Table 9: Reset Type Definitions*

Type	Description
Cold	Cold reset. The value is applied only at cold reset.
Fixed	The read value that applies at all times.

#### 1.4.4.12 Field Initialization

The Init keyword is optional and specifies an initialization recommendation for a register field.

If present, then there is an optional prefix that specifies the owner of the initialization. See Table 10 [Init Type Definitions].

- Example: Init: BIOS,2'b00. //A initialization recommendation for a field to be programmed by BIOS.

*Table 10: Init Type Definitions*

Type	Description
BIOS	Initialized by AMD provided AMD Generic Encapsulated Software Architecture (AGESA™) x86 software.
SBIOS	Initialized by OEM or IBV provided x86 software, also called Platform BIOS.
OS	Initialized by OS or Driver.

#### 1.4.4.13 Field Check

The Check keyword is optional and specifies the value that is recommended for firmware/software to write for a register field. It is a recommendation, not a requirement, and may not under all circumstances be what software programs.

#### 1.4.4.14 Field Valid Values

A register can optionally have either a valid values table or a valid bit table:

- A valid values table specifies the definition for specific field values.
- A valid bit table specifies the definition for specific field bits.

### 1.4.5 Revision History and Change Bar Notation

If a set of PDFs is generated to show differences with respect to a previous release, then:

- A revision history table is generated and exists before the Overview section. (highlighted in red in the following figure)
- The top line indicates what release the changes are in reference to.
- Changes in the revision history have 3 types:
  - Add. A heading, register or field is added.
  - Delete. A heading, register, or field is deleted.
  - Updated. A heading, register, or field is updated.

## Revision History

PPR Revision 3.05 Changes with respect to 3.02, Apr 9, 2024, PUB release:

- 1.2 [Reference Documents]: Updated.
- 1.4.3.2 [Physical Mnemonic]: Updated.
- 1.4.4.5.4 [BAR Access Method]: Updated.
- 1.4.4.5.4.1 [BAR as a Register Reference]: Updated.
- 1.4.4.5.6 [Data Port Access Method]: Updated.
- 1.8.1 [Features]: Updated.
- 1.8.2 [PCI Device ID Assignments]: Added.
- 2.1.1.1 [Core Definitions]: Updated.
- Core::X86::Apic::InterruptRequest: Updated.
- Core::X86::Apic::TriggerMode: Updated.
- Core::X86::Cpuid::CachePropEc3: Updated.
- Core::X86::Cpuid::ProcExtStateEnumEax00: Updated.
- Core::X86::Msr::CpuWdtCfg: Updated.
- Core::X86::Msr::MmioCfgBaseAddr: Updated.
- Core::X86::Pmc::Core::LsAnyFillsFromSys: Updated.
- Core::X86::Pmc::Core::LsDispatch: Updated.
- Core::X86::Pmc::Core::LsSmrRx: Updated.
- Core::X86::Pmc::Core::LsWebCloseFlush: Added.
- MCA::L5::MCA\_CTL\_MASK\_L5: Updated.
- MCA::IF::MCA\_CTL\_MASK\_IF: Updated.
- MCA::L2::MCA\_CTL\_MASK\_L2: Updated.
- MCA::FP::MCA\_CTL\_MASK\_FP: Updated.
- MCA::L3::MCA\_CTL\_MASK\_L3: Updated.
- MCA::L3::MCA\_IPID\_L3: Updated.
- MCA::CS::MCA\_IPID\_CS: Updated.
- MCA::PIE::MCA\_IPID\_PIE: Updated.
- MCA::UMC::MCA\_IPID\_UMC: Updated.
- 5.4.2.2.1 [SB-RMI Mailbox Sequence]: Updated.
- 5.6 [SB-RMI Registers]: Updated.
- 6.2.4 [Atomic Read Mechanism]: Updated.

### Bookmarks

- Legal Notices
- List of Chapters
- Table of Contents
- List of Figures
- List of Tables
- Revision History**
- 1 Overview
- 2 Core Complex (CCX)
- 3 Reliability, Availability, and Serviceability (RAS) Features
- 4 System Management Unit (SMU)
- 5 Advanced Platform Management Link (APML)
- 6 SB Temperature Sensor Interface (SB-TSI)
- List of Namespaces
- List of Definitions

Figure 19: Revision History Format Example

The following diagram shows the notation for changes:

- If a change exists on a line then a thin vertical bar is rendered in the left margin, as circled in blue.
- Deleted text is indicated with amber and strike-through, as circled in red.
- Added text is indicated with amber and underlined, as circled in green.

### 5.6 SB-RMI Registers

 Reads to unimplemented registers may return ~~00h~~ on zero value. Writes to unimplemented registers are discarded.

Figure 20: Change Notation Example

1.5     **Definitions**

*Table 11: Definitions*

Term	Description
<b>AGESA™</b>	AMD Generic Encapsulated Software Architecture.
<b>AP</b>	Application Processor.
<b>APML</b>	Advanced Platform Management Link.
<b>APU</b>	Accelerated Processing Unit.
<b>BatteryPower</b>	The system is running from a limited energy or battery power source or otherwise undocked from a continuous power supply. Setting using this definition may be required to change during run time.
<b>BCD</b>	Binary Coded Decimal number format.
<b>BCS</b>	Base Configuration Space.
<b>BIST</b>	Built-In Self-Test. Hardware within the processor that generates test patterns and verifies that they are stored correctly (in the case of memories) or received without error (in the case of links).
<b>Boot VID</b>	Boot Voltage ID. This is the VDD and VDDNB voltage level that the processor requests from the external voltage regulator during the initial phase of the cold boot sequence.
<b>C-states</b>	These are ACPI defined core power states. C0 is operational. All other C-states are low-power states in which the processor is not executing code. See docACPI.
<b>Cold reset</b>	PWROK is de-asserted and RESET_L is asserted.
<b>COF</b>	Current operating frequency of a given clock domain.
<b>DID</b>	Divisor Identifier. Specifies the post-PLL divisor used to reduce the COF.
<b>Doubleword</b>	A 32-bit value.
<b>DW</b>	Doubleword.
<b>EDC</b>	Electrical design current. Indicates the maximum current the voltage rail can demand for a short, thermally insignificant time.
<b>ECS</b>	Extended Configuration Space.
<b>FCH</b>	The integrated platform subsystem that contains the IO interfaces and bridges them to the system BIOS. Previously included in the Southbridge.
<b>FID</b>	Frequency Identifier. Specifies the PLL frequency multiplier for a given clock domain.
<b>FreeRunSampleTimer</b>	An internal free running timer used by many power management features.
<b>GB</b>	Gbyte or Gigabyte; 1,073,741,824 bytes.
<b>GT/s</b>	Giga-Transfers per second.
<b>HTC</b>	Hardware Thermal Control.
<b>HTC-active state</b>	Hardware-controlled lower-power, lower performance state used to reduce temperature.
<b>IFCM</b>	Isochronous flow-control mode, as defined in the link specification.
<b>IO configuration</b>	Access to configuration space through IO ports CF8h and CFCh.
<b>IP</b>	In electronic design, a semiconductor Intellectual Property, IP, or IP block is a reusable unit of logic, cell, or integrated circuit layout design that is the intellectual property of one party.
<b>KB</b>	Kbyte or Kilobyte; 1024 bytes.
<b>MB</b>	Megabyte; 1024 KB.
<b>MMIO</b>	Memory-Mapped Input-Output range. This is physical address space that is mapped to the IO functions such as the IO links or MMIO configuration.
<b>MMIO configuration</b>	Access to configuration space through memory space.
<b>OW</b>	Octword. An 128-bit value.
<b>PCIe®</b>	PCI Express.
<b>PCS</b>	Physical Coding Sublayer.
<b>Processor</b>	A package containing one or more Nodes. See Node.
<b>QW</b>	Quadword. A 64-bit value.
<b>REFCLK</b>	Reference clock. Refers to the clock frequency (100 MHz) or the clock period (10 ns) depending

	on the context used.
<b>RX</b>	Receiver.
<b>Shutdown</b>	A state in which the affected core waits for either INIT, RESET, or NMI. When shutdown state is entered, a shutdown special cycle is sent on the IO links.
<b>SMAF</b>	System Management Action Field. This is the code passed from the SMC to the processors in STPCLK assertion messages.
<b>SMC</b>	System Management Controller. This is the platform device that communicates system management state information to the processor through an IO link, typically the system IO hub.
<b>Speculative event</b>	A performance monitor event counter that counts all occurrences of the event even if the event occurs during speculative code execution.
<b>SSC</b>	Spread Spectrum Clocking.
<b>TDC</b>	Thermal Design Current.
<b>TDP</b>	Thermal Design Power. A power consumption parameter that is used in conjunction with thermal specifications to design appropriate cooling solutions for the processor.
<b>TOM</b>	Top of Memory.
<b>TOM2</b>	Top of extended Memory.
<b>TX</b>	Transmitter.
<b>UMI</b>	Unified Media Interface. The link between the processor and the FCH.
<b>VID</b>	Voltage level identifier.
<b>Warm reset</b>	RESET_L is asserted only (while PWROK stays high).
<b>XBAR</b>	Cross bar; command packet switch.

## 1.6 Changes Between Revisions and Product Variations

### 1.6.1 Revision Conventions

The processor revision is specified by CPUID\_Fn00000001\_EAX (FamModStep) or CPUID\_Fn80000001\_EAX (FamModStepExt). This document uses a revision letter instead of specific model numbers. Where applicable, the processor stepping is indicated after the revision letter. All behavior marked with a revision letter apply to future revisions unless they are superseded by a change in a later revision. See the revision guide in 1.2 [Reference Documents] for additional information about revision determination.

## 1.7 Package

### 1.7.1 Package type

The following packages are supported.

*Table 12: Package Definitions*

Term	Description
<b>FP8</b>	Notebook package for direct solder boards (uBGA).

## 1.8 Processor Overview

### 1.8.1 Features

The Family 1Ah Models 20h-2Fh Strix 1 Enthusiast, Gaming & Mainstream APU, Commercial APU microprocessor offering leadership CPU and GPU performance and battery life.

Product Features:

- CPU: Zen5
- GPU: Up to 8xWGP 4RB+ RDNA3.5
- NPU: AIE2+, 32TOPS (4x8)
- ISP: 2 MIPI Cameras
- PCIe®: x16Gen4
- DDR: 128b LP5/D5
- USB: 3x USB4
- Package: FP8

CCX Classic

- One Maurya CCX Complex, containing 4 Zen5 Classic CPU Core Instances
- Each Core may run in single-thread (1T) or SMT mode (2T), for up to 8T total
- 256-bit floating point
- 1MB of L2 Cache per CPU, for a total of 4MB L2
- 16MB L3 Cache, shared across all CPUs
- 1 SDP Port

CCX Dense

- One Maurya CCX Complex, containing 8 Zen5 Dense CPU Core Instances
- Each Core may run in single-thread (1T) or SMT mode (2T), for up to 16T total
- 256-bit floating point
- 1MB of L2 Cache per CPU, for a total of 8MB L2
- 8MB L3 Cache, shared across all CPUs
- 1 SDP Port

USB

- 3x USB Type-C (2x USB4 with DP Alt Mode and 1x USB3 Type-C)
- 1x USB 3.2 10 Gb/s
- 1x USB3 Secure Bio
- 3x USB 2.0 only

FCH

- ACPI
- CLKGEN/CGPLL
- GPIOs (S0 and S5)
- 4 general purpose I2C/I3C ports
  - Each port may be BIOS configurable at boot time as I2C or I3C
- I2C Slave Port for USB PD interface
- Real-Time Clock (RTC)
- 3x Quad SPI Ports (2 eSPI/SPI Combo ports, 1 Quad SPI port)
- UARTS (5 total controllers, max controller config is 1 4-wire interface and 4 2-wire interfaces)

## **2 Core Complex (CCX)**

### **2.1 Processor x86 Core**

#### **2.1.1 Core Functional Information**

##### **2.1.1.1 Core Definitions**

*Table 13: Definitions*



Term	Description
<b>BSC</b>	Boot strap core. Core 0 of the BSP.
<b>BSP</b>	Boot strap processor.
<b>Canonical-address</b>	An address in which the state of the most-significant implemented bit is duplicated in all the remaining higher-order bits, up to bit[63].
<b>CCX</b>	Core Complex where more than one core shares L3 resources.
<b>CMP</b>	Specifies the core number.
<b>Core</b>	The instruction execution unit of the processor when the term Core is used in a x86 core context.
<b>CoreCOF</b>	Core current operating frequency in MHz. CoreCOF = Core::X86::Msr::PStateDef[CpuFid[11:0]] * 5MHz.
<b>CPL</b>	Current Privilege Level of the running task when the term CPL is used in a x86 core context.
<b>CpuCoreNum</b>	Specifies the core number.
<b>#GP</b>	A general-protection exception.
<b>#GP(0)</b>	Notation indicating a general-protection exception (#GP) with error code of 0.
<b>HWPF</b>	Hardware Prefetcher.
<b>IBS</b>	Instruction based sampling.
<b>IO configuration</b>	Access to configuration space through IO ports CF8h and CFCh.
<b>IORR</b>	IO range register.
<b>L1 cache</b>	The level 1 caches (instruction cache and the data cache).
<b>L2 cache</b>	The level 2 caches.
<b>L3</b>	Level 3 Cache. The L3 term is also in Addrmaps to enumerate CCX units.
<b>L3 cache</b>	Level 3 Cache.
<b>Linear (virtual) address</b>	The address generated by a core after the segment is applied.
<b>LINT</b>	Local interrupt.
<b>Logical address</b>	The address generated by a core before the segment is applied.
<b>LRU</b>	Least recently used.
<b>LVT</b>	Local vector table. A collection of APIC registers that define interrupts for local events (e.g., APIC[530:500] [Extended Interrupt [3:0] Local Vector Table]).
<b>Macro-op</b>	The front-end of the pipeline breaks instructions into macro-ops and transfers (dispatches) them to the back-end of the pipeline for scheduling and execution. See Software Optimization Guide.
<b>Micro-op</b>	Processor schedulers break down macro-ops into sequences of even simpler instructions called micro-ops, each of which specifies a single primitive operation. See Software Optimization Guide.
<b>NBC</b>	NBC=(CPUID Fn00000001_EBX[LocalApicId[3:0]] == 0). Node Base Core. The lowest numbered core in the node.
<b>MTRR</b>	Memory-type range register. The MTRRs specify the type of memory associated with various memory ranges.
<b>MPB</b>	Microcode patch block.
<b>NTA</b>	Non-Temporal Access.
<b>PPIN</b>	Protected Processor Inventory Number.
<b>PTE</b>	Page table entry.
<b>SMI</b>	System management interrupt.
<b>SMM</b>	System Management Mode.
<b>SMT</b>	Simultaneous multithreading. See Core::X86::CpuId::CoreId[ThreadsPerCore].
<b>Speculative event</b>	A performance monitor event counter that counts all occurrences of the event even if the event occurs during speculative code execution.
<b>SVM</b>	Secure virtual machine.
<b>Thread</b>	One architectural context for instruction execution.

<b>VMPL</b>	Virtual Machine Privilege Level.
<b>WDT</b>	Watchdog timer. A timer that detects activity and triggers an error if a specified period of time expires without the activity.
<b>X2APICEN</b>	x2 APIC is enabled. X2APICEN = (Core::X86::Msr::APIC_BAR[ApicEn] && Core::X86::Msr::APIC_BAR[x2ApicEn]).

## 2.1.2 Secure Virtual Machine Mode (SVM)

Support for SVM mode is indicated by Core::X86::Cpuid::FeatureExtIdEcX[SVM].

### 2.1.2.1 BIOS support for SVM Disable

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

#### 2.1.2.1.1 Enable AMD Virtualization™

- Core::X86::Msr::VM\_CR[SvmeDisable] = 0.
- Core::X86::Msr::VM\_CR[Lock] = 1.
- Core::X86::Msr::SvmLockKey[SvmLockKey] = 0000\_0000\_0000\_0000h.

#### 2.1.2.1.2 Disable AMD Virtualization™

- Core::X86::Msr::SvmLockKey[SvmLockKey] = 0000\_0000\_0000\_0000h.
- Core::X86::Msr::VM\_CR[SvmeDisable] = 1.
- Core::X86::Msr::VM\_CR[Lock] = 1.

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

#### 2.1.2.1.3 Disable AMD Virtualization™, with a user supplied key

- Core::X86::Msr::VM\_CR[SvmeDisable] = 1.
- Core::X86::Msr::VM\_CR[Lock] = 1.
- Core::X86::Msr::SvmLockKey[SvmLockKey] programmed with value supplied by user. This value should be stored in NVRAM.

## 2.1.3 Microcode Patching

### 2.1.3.1 Microcode Patching Overview

The processor contains a small amount of RAM for implementing microcode patches. This patch RAM is loaded by a microcode routine (the Patch RAM Loader) that is part of the normal microcode contained in ROM. The patch RAM contains 64 microlines. Microlines are sets of processor microcode operations which are grouped to form a line of microcode. When the processor powers up, it uses its internal ROM microcode. If no patches are installed, the processor only executes microcode from the ROM.

The Patch RAM Loader function is called via a write to Core::X86::Msr::PATCH\_LOADER. The patch loader downloads microcode from the Microcode Patch Block (MPB) stored in memory into the processor patch RAM.

### 2.1.3.2 Microcode Patch Block (MPB)

The MPB is 14368 bytes in length and consists of two parts. The first part is a 32-byte header. The second part contains code, including the encrypted patch data. The properties of the patch allow for multiple consistency checks after decryption. If any consistency check fails, a #GP is generated. The patch data is encrypted to prevent unauthorized use of the patch mechanism.

Table 14: MPB Definition

Field	Byte Offset	Byte Length	Description
MPB_DATE	0000h	4	A doubleword for "mmddyyyy" in BCD format
Patch Level	0004h	4	A unique patch level as defined by the Revision Guide
MPB_LOADER_ID	0008h	2	A unique ID used for checking if the update is successful after the microcode patch function is executed.
MPB_SIZE	000Ah	2	
Reserved	000Ch	4	
MPB_NB_ID	0010h	4	The BIOS must match the chipset 1 device ID with this field before executing the microcode patch. A "0" means the patch is not chipset specific.
MPB_SB_ID	0014h	4	The BIOS must match the chipset 2 device ID with this field before executing the microcode patch. A "0" means the patch is not chipset specific.
MPB_REVISION	0018h	4	Microcode Patch Equivalent Processor Id.
MPB_BIOS_REVISION	001Ch	1	
Load Control	001Dh	1	Bit[0]: SerializedLoad: =0 (default), =1 The patch is required to be loaded serially across all cores in the system. Bit[1]: LoadOnBothThreads: = 0 (default) The patch is required to be loaded on both threads on each core in the system. =1 The patch may be loaded on one thread on each core in the system. Bits[7:2]: Reserved.
NEXT_MPB_OFFSET	001Eh	2	For MPB chaining. Offset of the next MPB in multiples of 16byte. An offset =0 indicates that no other MPB follows the current MPB.
Patch Data	0020h	14336	Patch Data.
Total MPB Size		14368	

### 2.1.4 Effective Frequency

The effective frequency interface allows software to discern the average, or effective, frequency of a given core over a configurable window of time. This provides software a measure of actual performance rather than forcing software to assume the current frequency of the core is the frequency of the last P-state requested. Core::X86::Msrr::MPERF is incremented by hardware at the P0 frequency while the core is in C0. Core::X86::Msrr::APERF increments in proportion to the actual number of core clocks cycles while the core is in C0.

The following procedure calculates effective frequency using Core::X86::Msrr::MPERF and Core::X86::Msrr::APERF:

1. At some point in time, write 0 to both MSRs.
2. At some later point in time, read both MSRs.
3. Effective frequency = (value read from Core::X86::Msrr::APERF / value read from Core::X86::Msrr::MPERF) \* P0 frequency.

## Additional notes:

- The amount of time that elapses between steps 1 and 2 is determined by software.
- It is software's responsibility to disable interrupts or any other events that may occur in between the Write of Core::X86::Msr::MPERF and the Write of Core::X86::Msr::APERF in step 1 or between the Read of Core::X86::Msr::MPERF and the Read of Core::X86::Msr::APERF in step 2.
- The behavior of Core::X86::Msr::MPERF and Core::X86::Msr::APERF may be modified by Core::X86::Msr::HWCR[EffFreqCntMwait].
- The effective frequency interface provides +/- 50MHz accuracy if the following constraints are met:
  - Effective frequency is read at most one time per millisecond.
  - When reading or writing Core::X86::Msr::MPERF and Core::X86::Msr::APERF software executes only MOV instructions, and no more than 3 MOV instructions, between the two RDMSR or WRMSR instructions.
  - Core::X86::Msr::MPERF and Core::X86::Msr::APERF are invalid if an overflow occurs.

## 2.1.5 Address Space

### 2.1.5.1 Virtual Address Space

The processor supports 48-bit address bits of virtual memory space (256 TB) as indicated by Core::X86::Cpuid::LongModeInfo.

### 2.1.5.2 Physical Address Space

The processor supports a 48-bit physical address space. See Core::X86::Cpuid::LongModeInfo.  
The processor issues a Decode Error for the following upper-address transactions (to address PhysAddr):

- Link or core requests with non-zero PhysAddr[63:48].

### 2.1.5.3 System Address Map

The processor defines a Reserved memory address region starting at FFFD\_0000\_0000h and extending up to FFFF\_FFFF\_FFFFh. System software must not map memory into this region. Downstream host accesses to the Reserved address region results in a page fault. Upstream system device accesses to the reserved address region results in an undefined operation.

#### 2.1.5.3.1 Memory Access to the Physical Address Space

All memory accesses to the physical address space from a core are sent to its associated Data Fabric (DF). All memory accesses from a link are routed through the DF. An IO link access to physical address space indicates to the DF the cache attribute (Coherent or Non-coherent, based on bit[0] of the Sized Read and Write commands).

A core access to physical address space has two important attributes that must be determined before issuing the access to the Northbridge: the memory type (e.g., WB, WC, UC; as described in the MTRRs) and the access destination (DRAM or MMIO).

If the memory map maps a region as DRAM that is not populated with real storage behind it, then that area of DRAM must be mapped as UC memtype.

This mechanism is managed by the BIOS and does not require any setup or changes by system software.

### 2.1.5.3.1.1 Determining Memory Type

The memory type for a core access is determined by the highest priority of the following ranges that the access falls in: 1=Lowest priority.

1. The memory type as determined by architectural mechanisms.
  - See the docAPM2 chapter titled "Memory System", sections "Memory-Type Range Registers" and "Page-Attribute Table Mechanism".
  - See the docAPM2 chapter titled "Nested Paging", section "Combining Memory Types, MTRRs".
  - See Core::X86::Msrr::MTRRdefType, Core::X86::Msrr::MtrrVarBase, Core::X86::Msrr::MtrrVarMask, Core::X86::Msrr::MtrrFix\_64K and Core::X86::Msrr::MtrrFix\_16K\_0 through Core::X86::Msrr::MtrrFix\_4K\_7.
2. TSeg & ASeg SMM mechanism (See Core::X86::Msrr::SMMAddr and Core::X86::Msrr::SMMMask).
3. CR0[CD]: If (CR0[CD] == 1) then MemType = CD.
4. MMIO configuration space, APIC space.
  - MMIO APIC space and MMIO config space must not overlap.
  - MemType = UC.
5. If ("In SMM Mode"&& ~((Core::X86::Msrr::SMMMask[AValid] && "The address falls within the ASeg region") || (Core::X86::Msrr::SMMMask[TValid] && "The address falls within the TSeg region")) then MemType = CD.

### 2.1.6 Configuration Space

PCI-defined configuration space was originally defined to allow up to 256 bytes of register space for each function of each device; these first 256 bytes are called base configuration space (BCS). It was expanded to support up to 4096 bytes per function; bytes 256 through 4095 are called extended configuration space (ECS).

The processor includes configuration space registers located in both BCS and ECS. Processor configuration space is accessed through bus 0, devices 18h to 1Fh, where device 18h corresponds to node 0 and device 1Fh corresponds to node 7. See 2.1.6.3 [Processor Configuration Space].

Configuration space is accessed by the processor through two methods as follows:

- IO-space configuration: IO instructions to addresses CF8h and CFCh.
  - Enabled through IO::IoCfgAddr[ConfigEn], which allows access to BCS.
  - Use of IO-space configuration can be programmed to generate GP faults through Core::X86::Msrr::HWCR[IoCfgGpFault].
  - SMI trapping for these accesses is specified by Core::X86::Msrr::SMI\_ON\_IO\_TRAP\_CTL\_STS and Core::X86::Msrr::SMI\_ON\_IO\_TRAP.
- Memory Mapped IO (MMIO) configuration: configuration space is a region of memory space.
  - The base address and size of this range is specified by Core::X86::Msrr::MmioCfgBaseAddr. The size is controlled by the number of configuration-space bus numbers supported by the system. Accesses to this range are converted configuration space as follows:
- Address[31:0] = {0h, segment[6:0], bus[7:0], device[4:0], function[2:0], offset[11:0]}.

The BIOS may use either configuration space access mechanism during boot. Before booting the OS, BIOS must disable IO access to ECS, enable MMIO configuration and build an ACPI defined MCFG table. BIOS ACPI code must use MMIO to access configuration space.

#### 2.1.6.1 Memory Mapped IO (MMIO) Configuration Coding Requirements

MMIO configuration space accesses must use the uncacheable (UC) memory type.

Instructions used to read MMIO configuration space are required to take the following form:

```
mov eax/ax/al, any_address_mode;
```

Instructions used to write MMIO configuration space are required to take the following form:

```
mov any_address_mode, eax/ax/al;
```

No other source/target registers may be used other than eax/ax/al.

In addition, all such accesses are required not to cross any naturally aligned DW boundary. Access to MMIO configuration space registers that do not meet these requirements result in undefined behavior.

### **2.1.6.2 MMIO Configuration Ordering**

Since MMIO configuration cycles are not serializing in the way that IO configuration cycles are, their ordering rules relative to posted may result in unexpected behavior.

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

### **2.1.6.3 Processor Configuration Space**

Accesses to unimplemented registers of implemented functions are ignored: Writes dropped; Reads return 0. Accesses to unimplemented functions also ignored: Writes are dropped; however, Reads return all F's. The processor does not log any Decode Error events for accesses to unimplemented registers or functions.

Accesses to device numbers of devices not implemented in the processor are routed based on the configuration map registers. If such access attempts cause Decode Errors, then the processor can log the event.

### **2.1.7 PCI Configuration Legacy Access**

**IOx0CF8 [IO-Space Configuration Address] (IO::IoCfgAddr)**

Read-write. Reset: 0000\_0000h.

IO::IoCfgAddr, and IO::IoCfgData are used to access system configuration space, as defined by the PCI specification. IO::IoCfgAddr provides the address register and IO::IoCfgData provides the data port. Software sets up the configuration address by writing to IO::IoCfgAddr. Then, when an access is made to IO::IoCfgData, the processor generates the corresponding configuration access to the address specified in IO::IoCfgAddr. See 2.1.6 [Configuration Space].

IO::IoCfgAddr may only be accessed through aligned, DW IO Reads and Writes; otherwise, the accesses are passed to the appropriate IO link. Accesses to IO::IoCfgAddr and IO::IoCfgData received from an IO link are treated as all other IO transactions received from an IO link. IO::IoCfgAddr and IO::IoCfgData in the processor are not accessible from an IO link.

\_aliasIO; IOx0CF8; IO=0000\_0000h

Bits	Description
31	<b>ConfigEn: configuration space enable.</b> Read-write. Reset: 0. 0=IO Read and Write accesses are passed to the appropriate IO link and no configuration access is generated. 1=IO Read and Write accesses to IO::IoCfgData are translated into configuration cycles at the configuration address specified by this register.
30:28	Reserved.
27:24	<b>ExtRegNo: extended register number.</b> Read-write. Reset: 0h. ExtRegNo provides bits[11:8] and RegNo provides bits[7:2] of the byte address of the configuration register.
23:16	<b>BusNo: bus number.</b> Read-write. Reset: 00h. Specifies the bus number of the configuration cycle.
15:11	<b>Device: device number.</b> Read-write. Reset: 00h. Specifies the device number of the configuration cycle.
10:8	<b>Function.</b> Read-write. Reset: 0h. Specifies the function number of the configuration cycle.
7:2	<b>RegNo: register address.</b> Read-write. Reset: 00h. See IO::IoCfgAddr[ExtRegNo].
1:0	Reserved.

**IOx0CFC [IO-Space Configuration Data Port] (IO::IoCfgData)**

Read-write. Reset: 0000\_0000h.

\_aliasIO; IOx0CFC; IO=0000\_0000h

Bits	Description
31:0	<b>Data.</b> Read-write. Reset: 0000_0000h. See IO::IoCfgAddr.

**2.1.8 System Software Interaction With SMT Enabled**

If Core::X86::Cpuid::CoreId[ThreadsPerCore] > 0, then SMT is enabled in all cores in the system. When SMT is enabled, the resources of each core are dynamically balanced among the hardware threads executing on that core. The number of hardware threads (hereafter "threads") supported by a single core when SMT is enabled is reported in Core::X86::Cpuid::CoreId[ThreadsPerCore]. System software that is SMT-aware may take advantage of the knowledge that core resources are being shared among multiple threads when scheduling tasks to be run by each thread on each core. System software that is not SMT-aware sees each thread as an independent core.



## 2.1.9 Register Sharing

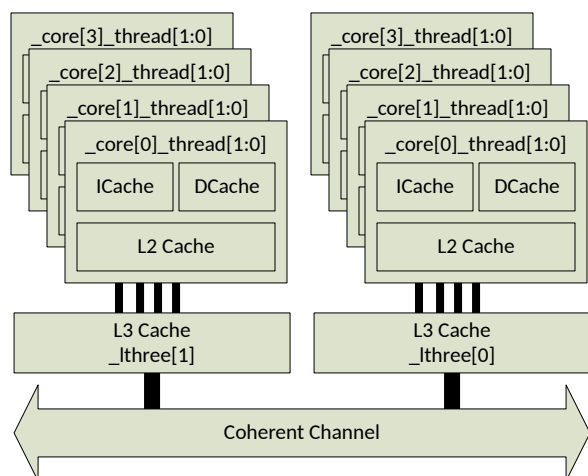


Figure 21: Register Sharing Domains

### MSR0000\_0010 [Time Stamp Counter] (TSC)

Read-write, Volatile. Reset: 0000 0000 0000 0000h.	
Core::X86::Msr::TSC_ithree[1:0]_core[3:0]_thread[1:0] MSR00000010	
Bits	Description
63:0	<b>TSC: time stamp counter.</b> Read-write, Volatile. Reset: 0. The TSC increments at the P0 frequency. The TSC counts at the same rate in all P-states, all C states, S0, or S1. A read of this MSR in guest mode is affected by Core::X86::Msr::TscRateMsr. The value (TSC/TSCRatio) is the TSC P0 frequency based

Figure 22: Instance Parameters

Instances of core registers are designated as lthree[n:0]\_core[n:0]\_thread[1:0]. Core registers may be shared at various levels of hierarchy as one register instance per node, per L3 complex, per core or per thread. The absence of the instance parameter \_thread[1:0] signifies that there is not a specific instance of said register per thread and thus the register is shared between thread[1] and thread[0]. Similarly, the absence of the instance parameter \_core[n:0] signifies that there is not a specific instance of said register per core and thus the register is shared by all cores in that L3 complex, and so on. The absence of instance parameters indicate there is one shared register at the node level. Software must coordinate writing to shared registers with other threads in the same sharing hierarchy level.

## 2.1.10 Timers

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

- Core::X86::Msr::TSC; the TSC increments at the rate specified by the P0 Pstate.
- The APIC timer (Core::X86::Apic::TimerInitialCount and Core::X86::Apic::TimerCurrentCount), which increments at the rate of 2xCLKIN; the APIC timer may increment in units of between 1 and 8.

## 2.1.11 Interrupts



### 2.1.11.1 System Management Mode (SMM)

System management mode (SMM) is typically used for system control activities such as power management. These activities are typically transparent to the operating system.

#### 2.1.11.1.1 SMM Overview

SMM is entered by a core on the next instruction boundary after a system management interrupt (SMI) is received and recognized. A core may be programmed to broadcast a special cycle to the system, indicating that it is entering SMM mode. The core then saves its state into the SMM memory state save area and jumps to the SMI service routine (or SMI handler). The pointer to the SMI handler is specified by MSRs. The code and data for the SMI handler are stored in the SMM memory area, which may be isolated from the main memory accesses.

The core returns from SMM by executing the RSM instruction from the SMI handler. The core restores its state from the SMM state save area and resumes execution of the instruction following the point where it entered SMM. The core may be programmed to broadcast a special bus cycle to the system, indicating that it is exiting SMM mode.

#### 2.1.11.1.2 Mode and Default Register Values

The software environment after entering SMM has the following characteristics:

- Addressing and operation is in Real mode.
  - A far jump, call or return in the SMI handler can only address the lower 1M of memory, unless the SMI handler first switches to protected mode.
  - If (Core::X86::Msr::SMM\_BASE[SmmBase] >= 0010\_0000h) then:
    - The value of the CS selector is undefined upon SMM entry.
    - The undefined CS selector value should not be used as the target of a far jump, call, or return.
- 4-Gbyte segment limits.
- Default 16-bit operand, address, and stack sizes (instruction prefixes can override these defaults).
- Control transfers that do not override the default operand size truncate the EIP to 16 bits.
- Far jumps or calls cannot transfer control to a segment with a base address requiring more than 20 bits, as in Real mode segment-base addressing, unless a change is made into protected mode.
- Interrupt vectors use the Real mode interrupt vector table.
- The IF flag in EFLAGS is cleared (INTR is not recognized).
- The TF flag in EFLAGS is cleared.
- The NMI and INIT interrupts are masked.
- Debug register DR7 is cleared (debug traps are disabled).

The SMM base address is specified by Core::X86::Msr::SMM\_BASE[SmmBase]. Important offsets to the base address pointer are:

- Core::X86::Msr::SMM\_BASE[SmmBase] + 8000h: SMI handler entry point.
- Core::X86::Msr::SMM\_BASE[SmmBase] + FE00h - FFFFh: SMM state save area.

#### 2.1.11.1.3 SMI Sources And Delivery

The processor accepts SMIs as link-defined interrupt messages only. The core/node destination of these SMIs is a function of the destination field of these messages. However, the expectation is that all such SMI messages are specified to be delivered globally (to all cores of all nodes).

There are also several local events that can trigger SMIs. However, these local events do not generate SMIs directly. Each of them triggers a programmable IO cycle that is expected to target the SMI command port in the IO hub and trigger a global SMI interrupt message back to the coherent fabric.

Local sources of SMI events that generate the IO cycle specified in Core::X86::Msr::SmiTrigIoCycle are:

- In the core, as specified by:
  - Core::X86::Msr::McExcepRedir.
  - Core::X86::Msr::SMI\_ON\_IO\_TRAP.
- All local APIC LVT registers programmed to generate SMIs.

The status for these are stored in Core::X86::Smm::LocalSmiStatus.

#### 2.1.11.1.4 SMM Initial State

After storing the save state, execution starts at Core::X86::Msr::SMM\_BASE[SmmBase] + 08000h. The SMM initial state is specified in the following table.

Table 15: SMM Initial State

Register	SMM Initial State
CS	SmmBase[19:4]
DS	0000h
ES	0000h
FS	0000h
GS	0000h
SS	0000h
General-Purpose Registers	Unmodified.
EFLAGS	0000_0002h
RIP	0000_0000_0000_8000h
CR0	Bits[0,2,3,31] cleared (PE, EM, TS, and PG); remainder is unmodified.
CR4	0000_0000_0000_0000h
GDTR	Unmodified.
LDTR	Unmodified.
IDTR	Unmodified.
TR	Unmodified.
DR6	Unmodified.
DR7	0000_0000_0000_0400h
EFER	All bits are cleared except bit[12] (SVME) which is unmodified.

#### 2.1.11.1.5 SMM Save State

In the following table, the offset field provides the offset from the SMM base address specified by Core::X86::Msr::SMM\_BASE[SmmBase].

Table 16: SMM Save State

Offset	Size	Contents	Access
FE00h	Word	ES Selector	Read-only
FE02h	6 Bytes		
FE08h	Quadword		
FE10h	Word	CS Selector	Read-only
FE12h	6 Bytes		
FE18h	Quadword		

FE20h	Word	SS	Selector	Read-only
FE22h	6 Bytes		Reserved	
FE28h	Quadword		Descriptor in memory format	
FE30h	Word	DS	Selector	Read-only
FE32h	6 Bytes		Reserved	
FE38h	Quadword		Descriptor in memory form	
FE40h	Word	FS	Selector	Read-only
FE42h	2 Bytes		Reserved	
FE44h	Doubleword		FS Base {16'b[47], 47:32}(note 1)	
FE48h	Quadword		Descriptor in memory format	
FE50h	Word	GS	Selector	Read-only
FE52h	2 Bytes		Reserved	
FE54h	Doubleword		GS Base {16'b[47], 47:32}(note 1)	
FE58h	Quadword		Descriptor in memory format	
FE60h	4 Bytes	GDTR	Reserved	Read-only
FE64h	Word		Limit	
FE66h	2 Bytes		Reserved	
FE68h	Quadword		Descriptor in memory format	
FE70h	Word	LDTR	Selector	Read-only
FE72h	Word		Attributes	
FE74h	Doubleword		Limit	
FE78h	Quadword		Base	
FE80h	4 Bytes	IDTR	Reserved	Read-only
FE84h	Word		Limit	
FE86h	2 Bytes		Reserved	
FE88h	Quadword		Base	
FE90h	Word	TR	Selector	Read-only
FE92h	Word		Attributes	
FE94h	Doubleword		Limit	
FE98h	Quadword		Base	
FEA0h	Quadword	IO_RESTART_RIP		
FEA8h	Quadword	IO_RESTART_RCX		
FEB0h	Quadword	IO_RESTART_RSI		
FEB8h	Quadword	IO_RESTART_RDI		
FEC0h	Doubleword	Core::X86::Smm::TrapOffset [SMM IO Trap Offset]		Read-only
FEC4	Doubleword	Core::X86::Smm::LocalSmiStatus		Read-only
FEC8h	Byte	Core::X86::Smm::IoRestart		Read-write
FEC9h	Byte	Core::X86::Smm::AutoHalt		Read-write
FECAh	Byte	Core::X86::Smm::NmiMask		Read-write
FECBh	5 Bytes	Reserved		
FED0h	Quadword	EFER		Read-only
FED8h	Quadword	Core::X86::Smm::SvmState		Read-only
FEE0h	Quadword	Guest VMCB physical address		Read-only

FEE8h	Quadword	SVM Virtual Interrupt Control	Read-only
FEF0h	16 Bytes	Reserved	
FEFCh	Doubleword	Core::X86::Smm::SmmRevID	Read-only
FF00h	Doubleword	Core::X86::Smm::SmmBase	Read-write
FF04h	28 Bytes	Reserved	
FF20h	Quadword	Guest PAT	Read-only
FF28h	Quadword	Host EFER (note 2)	
FF30h	Quadword	Host CR4 (note 2)	
FF38h	Quadword	Nested CR3 (note 2)	
FF40h	Quadword	Host CR0 (note 2)	
FF48h	Quadword	CR4	
FF50h	Quadword	CR3	
FF58h	Quadword	CR0	
FF60h	Quadword	DR7	Read-only
FF68h	Quadword	DR6	
FF70h	Quadword	RFLAGS	Read-write
FF78h	Quadword	RIP	Read-write
FF80h	Quadword	R15	
FF88h	Quadword	R14	
FF90h	Quadword	R13	
FF98h	Quadword	R12	
FFA0h	Quadword	R11	
FFA8h	Quadword	R10	
FFB0h	Quadword	R9	
FFB8h	Quadword	R8	
FFC0h	Quadword	RDI	Read-write
FFC8h	Quadword	RSI	
FFD0h	Quadword	RBP	
FFD8h	Quadword	RSP	
FFE0h	Quadword	RBX	
FFE8h	Quadword	RDX	
FFF0h	Quadword	RCX	
FFF8h	Quadword	RAX	
Notes: 1. This notation specifies that bit[47] is replicated in each of the 16 MSBs of the DW (sometimes called sign extended). The 16 LSBs contain bits[47:32]. 2. Only used for an SMI in guest mode with nested paging enabled.			

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

#### 2.1.11.1.6 System Management State

The following are offsets in the SMM save state area.

**SMMxFEC0 [SMM IO Trap Offset] (Core::X86::Smm::TrapOffset)**

Read-only, Volatile. Reset: 0000\_0000h.

If the assertion of SMI is recognized on the boundary of an IO instruction, Core::X86::Smm::TrapOffset contains information about that IO instruction. For example, if an IO access targets an unavailable device, the system can assert SMI and trap the IO instruction. Core::X86::Smm::TrapOffset then provides the SMI handler with information about the IO instruction that caused the trap. After the SMI handler takes the appropriate action, it can reconstruct and then re-execute the IO instruction from SMM. Or, more likely, it can use Core::X86::Smm::IoRestart to cause the core to re-execute the IO instruction immediately after resuming from SMM.

Bits	Description
31:16	<b>Port: trapped IO port address.</b> Read-only, Volatile. Reset: 0000h. This provides the address of the IO instruction.
15:12	<b>BPR: IO breakpoint match.</b> Read-only, Volatile. Reset: 0h.
11	<b>TF: EFLAGS TF value.</b> Read-only, Volatile. Reset: 0.
10:7	Reserved.
6	<b>SZ32: size 32 bits.</b> Read-only, Volatile. Reset: 0. 1=Port access was 32 bits.
5	<b>SZ16: size 16 bits.</b> Read-only, Volatile. Reset: 0. 1=Port access was 16 bits.
4	<b>SZ8: size 8 bits.</b> Read-only, Volatile. Reset: 0. 1=Port access was 8 bits.
3	<b>REP: repeated port access.</b> Read-only, Volatile. Reset: 0.
2	<b>STR: string-based port access.</b> Read-only, Volatile. Reset: 0.
1	<b>V: IO trap word valid.</b> Read-only, Volatile. Reset: 0. 0=The other fields of this offset are not valid. 1=The core entered SMM on an IO instruction boundary; all information in this offset is valid.
0	<b>RW: port access type.</b> Read-only, Volatile. Reset: 0. 0=IO Write (OUT instruction). 1=IO Read (IN instruction).

**SMMxFEC4 [Local SMI Status] (Core::X86::Smm::LocalSmiStatus)**

Read-only, Volatile. Reset: 0000\_0000h.

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

Bits	Description
31:9	Reserved.
8	<b>MceRedirSts: machine check exception redirection status.</b> Read-only, Volatile. Reset: 0. This bit is associated with the SMI source specified in Core::X86::Msr::McExcepRedir[RedirSmiEn].
7:4	Reserved.
3:0	<b>IoTrapSts: IO trap status.</b> Read-only, Volatile. Reset: 0h. Each of these bits is associated with each of the respective SMI sources specified in Core::X86::Msr::SMI_ON_IO_TRAP.

**SMMxFEC8 [IO Restart Byte] (Core::X86::Smm::IoRestart)**

Read-write. Reset: 00h.

If the core entered SMM on an IO instruction boundary, the SMI handler may write this to FFh. This causes the core to re-execute the trapped IO instruction immediately after resuming from SMM. The SMI handler should only write to this byte if Core::X86::Smm::TrapOffset[V] == 1; otherwise, the behavior is undefined.

If a second SMI is asserted while a valid IO instruction is trapped by the first SMI handler, the core services the second SMI prior to re-executing the trapped IO instruction. Core::X86::Smm::TrapOffset[V] == 0 during the second entry into SMM, and the second SMI handler must not rewrite this byte.

If there is a simultaneous SMI IO instruction trap and debug breakpoint trap, the processor first responds to the SMI and postpones recognizing the debug exception until after resuming from SMM. If debug registers other than DR6 and DR7 are used while in SMM, they must be saved and restored by the SMI handler. If Core::X86::Smm::IoRestart is set to FFh when the RSM instruction is executed, the debug trap does not occur until after the IO instruction is re-executed.

Bits	Description
7:0	<b>RST: SMM IO Restart Byte.</b> Read-write. Reset: 00h.

**SMMxFEC9 [Auto Halt Restart Offset] (Core::X86::Smm::AutoHalt)**

Read-write. Reset: 00h.

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

**SMMxFECA [NMI Mask] (Core::X86::Smm::NmiMask)**

Read-write. Reset: 00h.

Bits	Description
7:1	Reserved.
0	<b>NmiMask: NMI Mask.</b> Read-write. Reset: 0. 0=NMI not masked. 1=NMI masked. Specifies whether NMI was masked upon entry to SMM.

**SMMxFED8 [SMM SVM State] (Core::X86::Smm::SvmState)**

Read-only, Volatile. Reset: 0000\_0000\_0000\_0000h.

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

Bits	Description														
63:4	Reserved.														
3	<b>HostEflagsIF: host EFLAGS IF.</b> Read-only, Volatile. Reset: 0.														
2:0	<b>SvmState.</b> Read-only, Volatile. Reset: 0h. <b>ValidValues:</b>														
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>SMM entered from a non-guest state.</td></tr> <tr> <td>1h</td><td>Reserved.</td></tr> <tr> <td>2h</td><td>SMM entered from a guest state.</td></tr> <tr> <td>5h-3h</td><td>Reserved.</td></tr> <tr> <td>6h</td><td>SMM entered from a guest state with nested paging enabled.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	SMM entered from a non-guest state.	1h	Reserved.	2h	SMM entered from a guest state.	5h-3h	Reserved.	6h	SMM entered from a guest state with nested paging enabled.	7h	Reserved.
Value	Description														
0h	SMM entered from a non-guest state.														
1h	Reserved.														
2h	SMM entered from a guest state.														
5h-3h	Reserved.														
6h	SMM entered from a guest state with nested paging enabled.														
7h	Reserved.														

**SMMxFEFC [SMM Revision Identifier] (Core::X86::Smm::SmmRevID)**

Read-only. Reset: 0003\_0064h.

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

Bits	Description
31:18	Reserved.
17	<b>BRL.</b> Read-only. Reset: 1. 1=Base relocation supported.
16	<b>IOTrap.</b> Read-only. Reset: 1. 1=IO trap supported.
15:0	<b>Revision.</b> Read-only. Reset: 0064h.

<b>SMMxFE00 [SMM Base Address] (Core::X86::Smm::SmmBase)</b>	
Read-write, Volatile. Reset: 0000_0000_0000_0000h.	
This offset stores the base of the SMM-State of the processor upon entry into SMM.	
Bits	Description
63:32	Reserved.
31:0	<b>SmmBase.</b> Read-write, Volatile. Reset: 0000_0000h. See Core::X86::Msrr::SMM_BASE[SmmBase].

#### 2.1.11.1.7 Exceptions and Interrupts in SMM

When SMM is entered, the core masks INTR, NMI, SMI, and INIT interrupts. The core clears the IF flag to disable INTR interrupts. To enable INTR interrupts within SMM, the SMM handler must set the IF flag to 1.

Generating an INTR interrupt can be used for unmasking NMI interrupts in SMM. The core recognizes the assertion of NMI within SMM immediately after the completion of an IRET instruction. Once NMI is recognized within SMM, NMI recognition remains enabled until SMM is exited, at which point NMI masking is restored to the state it was in before entering SMM.

While in SMM, the core responds to STPCLK interrupts, as well as to all exceptions that may be caused by the SMI handler.

#### 2.1.11.1.8 The Protected ASeg and TSeg Areas

These ranges are controlled by Core::X86::Msrr::SMMAddr and Core::X86::Msrr::SMMMask; see those registers for details.

#### 2.1.11.1.9 SMM Special Cycles

Special cycles can be initiated on entry and exit from SMM to acknowledge to the system that these transitions are occurring. These are controlled by Core::X86::Msrr::HWCR[RsmSpCycDis, SmiSpCycDis].

#### 2.1.11.1.10 Locking SMM

The SMM registers (Core::X86::Msrr::SMMAddr and Core::X86::Msrr::SMMMask) can be locked from being altered by setting Core::X86::Msrr::HWCR[SmmLock]. SBIOS must lock the SMM registers after initialization to prevent unexpected changes to these registers.

#### 2.1.11.1.11 SMM Page Configuration Lock

The SMM Page Configuration Lock feature allows SMM handler code to lock the paging configuration. Once locked, the paging configuration cannot be modified until RSM completes.

Core::X86::CpuId::FeatureExt2Eax[SmmPgCfgLock] Specifies SMM page configuration locking is supported.

Core::X86::Msrr::HWCR[SmmPgCfgLock] locks registers related to page configuration. If not in SMM mode, Error-on-write-1. Cleared on RSM instruction.

If Core::X86::Msrr::HWCR[SmmPgCfgLock], WRMSR of Core::X86::Msrr::EFER results in an error.

If Core::X86::Msrr::HWCR[SmmPgCfgLock], MOV CR0, CR3 and CR4 instructions result in an error.

#### 2.1.11.2 Local APIC

The processor supports the APIC interrupt controller and the X2APIC interrupt controllers.

See 2.1.11.2.2 [Local APIC Registers] for the APIC registers and Core::X86::Msrr::APIC\_ID through



Core::X86::Msr::ExtendedInterruptLvtEntries for the X2APIC registers.

#### 2.1.11.2.1 Local APIC Functional Description

The local APIC contains logic to receive interrupts from a variety of sources and to send interrupts to other local APICs, as well as registers to control its behavior and report status. Interrupts can be received from:

- IO devices including the IO hub (IO APICs)
- Other local APICs (inter-processor interrupts)
- APIC timer
- Thermal events
- Performance counters
- Legacy local interrupts from the IO hub (INTR and NMI)
- APIC internal errors

The APIC timer, thermal events, performance counters, local interrupts, and internal errors are all considered local interrupt sources, and their routing is controlled by local vector table entries. These entries assign a message type and vector to each interrupt, allow them to be masked, and track the status of the interrupt.

IO and inter-processor interrupts have their message type and vector assigned at the source and are unaltered by the local APIC. They carry a destination field and a mode bit that together determine which local APIC(s) accepts them. The destination mode (DM) bit specifies if the interrupt request packet should be handled in physical or logical destination mode.

##### 2.1.11.2.1.1 Detecting and Enabling

The presence of APIC is detected via Core::X86::Cpuid::FeatureIdEdx[APIC], and the presence of X2APIC is detected via Core::X86::Cpuid::FeatureIdEcX[X2APIC].

The local APIC is enabled via Core::X86::Msr::APIC\_BAR[ApicEn]. The X2APIC is enabled via Core::X86::Msr::APIC\_BAR[x2ApicEn]. Reset forces the APIC and X2APIC disabled.

##### 2.1.11.2.1.2 APIC Register Space

MMIO APIC space:

- Memory mapped to a 4-KB range. The memory type of this space is the UC memory type. The base address of this range is specified by {Core::X86::Msr::APIC\_BAR[ApicBar[47:12]],000h}.
- The mnemonic is defined to be APICxXXX; where XXX is the byte address offset from the base address starting with APICx020 through APICx530 (Core::X86::Apic::ApicId - Core::X86::Apic::ExtendedInterruptLvtEntries).
- Treated as normal memory space when APIC is disabled, as specified by Core::X86::Msr::APIC\_BAR[ApicEn].

MSR X2APIC space:

- The local APIC register space in x2APIC mode.
- MMIO APIC registers in x2APIC mode is defined by the register from MSR0000\_0802 to MSR0000\_08[53:50] (Core::X86::Msr::APIC\_ID through Core::X86::Msr::ExtendedInterruptLvtEntries).
- If (Core::X86::Msr::APIC\_BAR[x2ApicEn] == 0) then GP-read-write.
- RDMSR/WRMSR will occur in program order.

##### 2.1.11.2.1.3 ApicId Enumeration Requirements

Note: do not require contiguous ApicId assignments.

Operating systems are expected to use Core::X86::Cpuid::SizeId[ApicIdSize], the number of least significant bits in the Initial APIC ID that indicate core ID within a processor, in constructing per-core CPUID



masks. Core::X86::Cpuid::SizeId[ApicIdSize] determines the maximum number of cores (MNC) that the processor could theoretically support, not the actual number of cores that are actually implemented or enabled on the processor, as indicated by Core::X86::Cpuid::SizeId[NC].

#### 2.1.11.2.1.4 Physical Destination Mode

The interrupt is only accepted by the local APIC whose Core::X86::Apic::ApicId[ApicId] matches the destination field of the interrupt. Physical mode allows up to 255 APICs to be addressed individually.

#### 2.1.11.2.1.5 Logical Destination Mode

A local APIC accepts interrupts selected by Core::X86::Apic::LocalDestination and the destination field of the interrupt using either cluster or flat format as configured by Core::X86::Apic::DestinationFormat[Format].

If flat destinations are in use, bits[7:0] of Core::X86::Apic::LocalDestination[Destination] are checked against bits[7:0] of the arriving interrupt's destination field. If any bit position is set in both fields, the local APIC is a valid destination. Flat format allows up to 8 APICs to be addressed individually.

If cluster destinations are in use, bits[7:4] of Core::X86::Apic::LocalDestination[Destination] are checked against bits[7:4] of the arriving interrupt's destination field to identify the cluster. If all of bits[7:4] match, then bits[3:0] of Core::X86::Apic::LocalDestination[Destination] and the interrupt destination are checked for any bit positions that are set in both fields to identify processors within the cluster. If both conditions are met, the local APIC is a valid destination. Cluster format allows 15 clusters of 4 APICs each to be addressed.

#### 2.1.11.2.1.6 Interrupt Delivery

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

When an APIC accepts a non-vectored interrupt, it is handled directly by the processor instead of being queued in the APIC. When an APIC accepts a fixed or lowest-priority interrupt, it sets the bit in Core::X86::Apic::InterruptRequest corresponding to the vector in the interrupt. For local interrupt sources, this comes from the vector field in that interrupt's local vector table entry. The corresponding bit in Core::X86::Apic::TriggerMode is set if the interrupt is level-triggered and cleared if edge-triggered. If a subsequent interrupt with the same vector arrives when the corresponding bit in Core::X86::Apic::InterruptRequest[RequestBits] is already set, the two interrupts are collapsed into one. Vectors[15:0] are Reserved.

#### 2.1.11.2.1.7 Vectored Interrupt Handling

Core::X86::Apic::TaskPriority and Core::X86::Apic::ProcessorPriority each contain an 8-bit priority divided into a main priority (bits[7:4]) and a priority sub-class (bits[3:0]). The task priority is assigned by software to set a threshold priority at which the processor is interrupted.

The processor priority is calculated by comparing the main priority (bits[7:4]) of Core::X86::Apic::TaskPriority[Priority] to bits[7:4] of the 8-bit encoded value of the highest bit set in Core::X86::Apic::InService. The processor priority is the higher of the two main priorities.

The processor priority is used to determine if any accepted interrupts (indicated by Core::X86::Apic::InterruptRequest[RequestBits]) are high enough priority to be serviced by the processor. When the processor is ready to service an interrupt, the highest bit in Core::X86::Apic::InterruptRequest[RequestBits] is cleared, and the corresponding bit is set in Core::X86::Apic::InService[InServiceBits].

When the processor has completed service for an interrupt, it performs a Write to Core::X86::Apic::EndOfInterrupt, clearing the highest bit in Core::X86::Apic::InService[InServiceBits] and causing the next-highest interrupt to be serviced. If the corresponding bit in Core::X86::Apic::TriggerMode[TriggerModeBits] is set, a Write to Core::X86::Apic::EndOfInterrupt is performed on all APICs to complete service of the interrupt at the source.

#### 2.1.11.2.1.8 Interrupt Masking

Interrupt masking is controlled by the Core::X86::Apic::ExtendedApicControl. If Core::X86::Apic::ExtendedApicControl[IerEn] is set, Core::X86::Apic::InterruptEnable are used to mask interrupts. Any bit in Core::X86::Apic::InterruptEnable[InterruptEnableBits] that is clear indicates the corresponding interrupt is masked. A masked interrupt is not serviced and the corresponding bit in Core::X86::Apic::InterruptRequest[RequestBits] remains set.

#### 2.1.11.2.1.9 Spurious Interrupts

In the event that the task priority is set to or above the level of the interrupt to be serviced, the local APIC delivers a spurious interrupt vector to the processor, as specified by Core::X86::Apic::SpuriousInterruptVector. Core::X86::Apic::InService is not changed and no Write to Core::X86::Apic::EndOfInterrupt occurs.

#### 2.1.11.2.1.10 Spurious Interrupts Caused by Timer Tick Interrupt

A typical interrupt is asserted until it is serviced. An interrupt is de-asserted when software clears the interrupt status bit within the interrupt service routine. Timer tick interrupt is an exception since it is de-asserted regardless of whether it is serviced or not.

The processor is not always able to service interrupts immediately (i.e., when interrupts are masked by clearing EFLAGS.IM).

If the processor is not able to service the timer tick interrupt for an extended period of time, the INTR caused by the first timer tick interrupt asserted during that time is delivered to the local APIC in ExtInt mode and latched, and the subsequent timer tick interrupts are lost. The following cases are possible when the processor is ready to service interrupts:

- An ExtInt interrupt is pending, and INTR is asserted. This results in timer tick interrupt servicing. This occurs 50 percent of the time.
- An ExtInt interrupt is pending, and INTR is de-asserted. The processor sends the interrupt acknowledge cycle, but when the PIC receives it, INTR is de-asserted, and the PIC sends a spurious interrupt vector. This occurs 50 percent of the time.

There is a 50 percent probability of spurious interrupts to the processor.

#### 2.1.11.2.1.11 Lowest-Priority Interrupt Arbitration

Fixed and non-vectored interrupts are accepted by their destination APICs without arbitration.

Delivery of lowest-priority interrupts requires all APICs to arbitrate to determine which one accepts the interrupt. If Core::X86::Apic::SpuriousInterruptVector[FocusDisable] is clear, then the focus processor for an interrupt always accepts the interrupt. A processor is the focus of an interrupt if it is already servicing that interrupt (corresponding bit in Core::X86::Apic::InService[InServiceBits] is set) or if it already has a pending request for that interrupt (corresponding bit in Core::X86::Apic::InterruptRequest[RequestBits] is set). If Core::X86::Apic::ExtendedApicControl[IerEn] is set, the interrupt must also be enabled in Core::X86::Apic::InterruptEnable[InterruptEnableBits] for a processor to be the focus processor. If there is no focus processor for an interrupt, or focus processor checking is disabled, then each APIC calculates an arbitration priority value, stored in Core::X86::Apic::ArbitrationPriority, and the one with the lowest result

accepts the interrupt.

The arbitration priority value is calculated by comparing Core::X86::Apic::TaskPriority[Priority] with the 8-bit encoded value of the highest bit set in Core::X86::Apic::InterruptRequest[RequestBits] (IRRVec) and the 8-bit encoded value of the highest bit set Core::X86::Apic::InService[InServiceBits] (ISRVec). If Core::X86::Apic::ExtendedApicControl[IerEn] is set the IRRVec and ISRVec are based off the highest enabled interrupt. The main priority bits[7:4] are compared as follows:

```
if ((TaskPriority[Priority[7:4]] >= InterruptRequest[IRRVec[7:4]])
&&(TaskPriority[Priority[7:4]] > InService[ISRVec[7:4]]) {
ArbitrationPriority[Priority] = TaskPriority[Priority]
} elseif { (InterruptRequest[IRRVec[7:4]] > InService[ISRVec[7:4]])
ArbitrationPriority[Priority] = {InterruptRequest[IRRVec[7:4]], 0h}
} else {
ArbitrationPriority[Priority] = {InService[ISRVec[7:4]], 0h}
}
```

### 2.1.11.2.1.12 Inter-Processor Interrupts

The Core::X86::Apic::InterruptCommandLow and Core::X86::Apic::InterruptCommandHigh provide a mechanism for generating interrupts in order to redirect an interrupt to another processor, originate an interrupt to another processor, or allow a processor to interrupt itself. A Write to register Core::X86::Apic::InterruptCommandLow causes an interrupt to be generated with the properties specified by the Core::X86::Apic::InterruptCommandLow and Core::X86::Apic::InterruptCommandHigh fields.

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

Note: x indicates a don't care.

*Table 17: ICR Valid Combinations*

Message Type	Trigger Mode	Level	Destination Shorthand
Fixed	Edge	x	x
	Level	Assert	x
Lowest Priority, SMI, NMI, INIT	Edge	x	Destination or all excluding self
	Level	Assert	Destination or all excluding self
Startup	x	x	Destination or all excluding self

### 2.1.11.2.1.13 APIC Timer Operation

The local APIC contains a 32-bit timer, controlled by Core::X86::Apic::TimerLvtEntry, Core::X86::Apic::TimerInitialCount, and Core::X86::Apic::TimerDivideConfiguration. The processor bus clock is divided by the value in Core::X86::Apic::TimerDivideConfiguration[Div[3:0]] to obtain a time base for the timer. When Core::X86::Apic::TimerInitialCount[Count] is written, the value is copied into Core::X86::Apic::TimerCurrentCount. Core::X86::Apic::TimerCurrentCount[Count] is decremented at the rate of the divided clock. When the count reaches 0, a timer interrupt is generated with the vector specified in Core::X86::Apic::TimerLvtEntry[Vector]. If Core::X86::Apic::TimerLvtEntry[Mode] specifies periodic operation, Core::X86::Apic::TimerCurrentCount[Count] is reloaded with the Core::X86::Apic::TimerInitialCount[Count] value, and it continues to decrement at the rate of the

divided clock. If Core::X86::Apic::TimerLvtEntry[Mask] is set, timer interrupts are not generated.

#### 2.1.11.2.1.14 Generalized Local Vector Table

All LVTs (Core::X86::Apic::ThermalLvtEntry to Core::X86::Apic::LVTLINT, and Core::X86::Apic::ExtendedInterruptLvtEntries) support a generalized message type as follows:

- 000b=Fixed
- 010b=SMI
- 100b=NMI
- 111b=ExtINT
- All other messages types are Reserved.

#### 2.1.11.2.1.15 State at Reset

At power-up or reset, the APIC is hardware disabled (Core::X86::Msr::APIC\_BAR[ApicEn] == 0) so only SMI, NMI, INIT, and ExtInt interrupts may be accepted.

The APIC can be software disabled through Core::X86::Apic::SpuriousInterruptVector[APICSWEn]. The software disable has no effect when the APIC is hardware disabled.

When a processor accepts an INIT interrupt, the APIC is reset as at power-up, with the exception that:

- Core::X86::Apic::ApicId is unaffected.
- Pending APIC register writes complete.

#### 2.1.11.2.2 Local APIC Registers

APICx020 [APIC ID] (Core::X86::Apic::ApicId)	
Read-only.	
_lthree[1:0]_core[7:0]_thread[1:0]; APICx020; APIC={Core::X86::Msr::APIC_BAR[ApicBar[47:12]] , 000h}	
Bits	Description
31:24	<b>ApicId: APIC ID.</b> Read-only. Reset: XXh. The reset value varies based on core number. See 2.1.11.2.1.3 [ApicId Enumeration Requirements].
23:0	Reserved.
APICx030 [APIC Version] (Core::X86::Apic::ApicVersion)	
Read-only.	
_lthree[1:0]_core[7:0]_thread[1:0]; APICx030; APIC={Core::X86::Msr::APIC_BAR[ApicBar[47:12]] , 000h}	
Bits	Description
31	<b>ExtApicSpace: extended APIC register space present.</b> Read-only. Reset: 1. 1=Indicates the presence of extended APIC register space starting at Core::X86::Apic::ExtendedApicFeature.
30:25	Reserved.
24	<b>DirectedEoiSupport: directed EOI support.</b> Read-only. Reset: Fixed,1. 0=Directed EOI capability not supported.
23:16	<b>MaxLvtEntry.</b> Read-only. Reset: XXh. Specifies the number of entries in the local vector table minus one.
15:8	Reserved.
7:0	<b>Version.</b> Read-only. Reset: 10h. Indicates the version number of this APIC implementation.

**APICx080 [Task Priority] (Core::X86::Apic::TaskPriority)**

Read-write. Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx080; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

Bits	Description
31:8	Reserved.
7:0	<b>Priority.</b> Read-write. Reset: 00h. This field is assigned by software to set a threshold priority at which the core is interrupted.

**APICx090 [Arbitration Priority] (Core::X86::Apic::ArbitrationPriority)**

Read-only, Volatile. Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx090; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

Bits	Description
31:8	Reserved.
7:0	<b>Priority.</b> Read-only, Volatile. Reset: 00h. Indicates the current priority for a pending interrupt, or a task or interrupt being serviced by the core. The priority is used to arbitrate between cores to determine which accepts a lowest-priority interrupt request.

**APICx0A0 [Processor Priority] (Core::X86::Apic::ProcessorPriority)**

Read-only, Volatile. Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx0A0; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

Bits	Description
31:8	Reserved.
7:0	<b>Priority.</b> Read-only, Volatile. Reset: 00h. Indicates the core's current priority servicing a task or interrupt, and is used to determine if any pending interrupts should be serviced. It is the higher value of the task priority value and the current highest in-service interrupt.

**APICx0B0 [End of Interrupt] (Core::X86::Apic::EndOfInterrupt)**

Write-only.

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

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx0B0; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

Bits	Description
31:0	Reserved.

**APICx0C0 [Reserved] (Core::X86::Apic::RemoteRead)**

Read-only. Reset: 0000\_0000h.

Remote Read is not supported.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx0C0; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

Bits	Description
31:0	Reserved.

**APICx0D0 [Logical Destination] (Core::X86::Apic::LocalDestination)**

Read-write, Volatile. Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx0D0; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

Bits	Description
31:24	<b>Destination.</b> Read-write, Volatile. Reset: 00h. This APIC's destination identification. Used to determine which interrupts should be accepted.
23:0	Reserved.

**APICx0E0 [Destination Format] (Core::X86::Apic::DestinationFormat)**

Read-write. Reset: F000\_0000h.

Only supported in xAPIC mode.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx0E0; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

Bits	Description								
31:28	<b>Format.</b> Read-write. Reset: Fh. Controls which format to use when accepting interrupts with a logical destination mode. <b>ValidValues:</b>								
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Cluster destinations are used.</td></tr> <tr> <td>Eh-1h</td><td>Reserved.</td></tr> <tr> <td>Fh</td><td>Flat destinations are used.</td></tr> </table>	Value	Description	0h	Cluster destinations are used.	Eh-1h	Reserved.	Fh	Flat destinations are used.
Value	Description								
0h	Cluster destinations are used.								
Eh-1h	Reserved.								
Fh	Flat destinations are used.								
27:0	Reserved.								

**APICx0F0 [Spurious-Interrupt Vector] (Core::X86::Apic::SpuriousInterruptVector)**

Reset: 0000\_00FFh.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx0F0; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

Bits	Description
31:10	Reserved.
9	<b>FocusDisable.</b> Read-write. Reset: 0. 1=Disable focus core checking during lowest-priority arbitrated interrupts.
8	<b>APICSWEn: APIC software enable.</b> Read-write, Volatile. Reset: 0. 0=SMI, NMI, INIT, LINT[1:0], and Startup interrupts may be accepted; pending interrupts in Core::X86::Apic::InService and Core::X86::Apic::InterruptRequest are held, but further fixed, lowest-priority, and ExtInt interrupts are not accepted. All LVT entry mask bits are set and cannot be cleared.
7:0	<b>Vector.</b> Read-write, Volatile. Reset: FFh. The vector that is sent to the core in the event of a spurious interrupt.

**APICx1[0...7]0 [In-Service] (Core::X86::Apic::InService)**

Read-only, Volatile. Reset: 0000\_0000h.

The in-service registers provide a bit per interrupt to indicate that the corresponding interrupt is being serviced by the core. The first 16 InServiceBits of the first Core::X86::Apic::InService register are Reserved.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n0; APICx100; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n1; APICx110; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n2; APICx120; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n3; APICx130; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n4; APICx140; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n5; APICx150; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n6; APICx160; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n7; APICx170; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

Bits	Description
31:0	<b>InServiceBits.</b> Read-only, Volatile. Reset: 0000_0000h. These bits are set when the corresponding interrupt is being serviced by the core.

**APICx1[8...F]0 [Trigger Mode] (Core::X86::Apic::TriggerMode)**

Read-only, Volatile. Reset: 0000\_0000h.

The trigger mode registers provide a bit per interrupt to indicate the assertion mode of each interrupt. The first 16 TriggerModeBits of the each thread's APIC[1F0:180] registers are Reserved.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n0; APICx180; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n1; APICx190; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n2; APICx1A0; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n3; APICx1B0; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n4; APICx1C0; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n5; APICx1D0; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n6; APICx1E0; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n7; APICx1F0; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

Bits	Description
31:0	<b>TriggerModeBits.</b> Read-only, Volatile. Reset: 0000_0000h. The corresponding trigger mode bit is updated when an interrupt is accepted. 1=Level-triggered interrupt. 0=Edge-triggered interrupt. <b>ValidValues:</b> See: Core::X86::Apic::TriggerMode[TriggerModeBits] Valid Values.

*Core::X86::Apic::TriggerMode[TriggerModeBits] Valid Values*

Bit	Description
[0]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[1]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[2]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[3]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[4]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[5]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[6]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[7]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[8]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[9]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[10]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[11]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[12]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[13]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[14]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[15]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[16]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[17]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[18]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[19]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[20]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[21]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[22]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[23]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[24]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[25]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[26]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[27]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[28]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[29]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
[30]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.



[31]	0=Edge-triggered interrupt. 1=Level-triggered interrupt.
<b>APICx2[0...7]0 [Interrupt Request] (Core::X86::Apic::InterruptRequest)</b>	
Read-only. Reset: 0000_0000h.	
The interrupt request registers provide a bit per interrupt to indicate that the corresponding interrupt has been accepted by the APIC. The first 16 RequestBits of the first Core::X86::Apic::InterruptRequest register are Reserved.	
_lthree[1:0]_core[7:0]_thread[1:0]_n0; APICx200; APIC={Core::X86::Msr::APIC_BAR[ApicBar[47:12]] , 000h}	
_lthree[1:0]_core[7:0]_thread[1:0]_n1; APICx210; APIC={Core::X86::Msr::APIC_BAR[ApicBar[47:12]] , 000h}	
_lthree[1:0]_core[7:0]_thread[1:0]_n2; APICx220; APIC={Core::X86::Msr::APIC_BAR[ApicBar[47:12]] , 000h}	
_lthree[1:0]_core[7:0]_thread[1:0]_n3; APICx230; APIC={Core::X86::Msr::APIC_BAR[ApicBar[47:12]] , 000h}	
_lthree[1:0]_core[7:0]_thread[1:0]_n4; APICx240; APIC={Core::X86::Msr::APIC_BAR[ApicBar[47:12]] , 000h}	
_lthree[1:0]_core[7:0]_thread[1:0]_n5; APICx250; APIC={Core::X86::Msr::APIC_BAR[ApicBar[47:12]] , 000h}	
_lthree[1:0]_core[7:0]_thread[1:0]_n6; APICx260; APIC={Core::X86::Msr::APIC_BAR[ApicBar[47:12]] , 000h}	
_lthree[1:0]_core[7:0]_thread[1:0]_n7; APICx270; APIC={Core::X86::Msr::APIC_BAR[ApicBar[47:12]] , 000h}	
<b>Bits</b>	<b>Description</b>
31:0	<b>RequestBits.</b> Read-only. Reset: 0000_0000h. The corresponding request bit is set when the an interrupt is accepted by the APIC.
	<b>ValidValues:</b> See: Core::X86::Apic::InterruptRequest[RequestBits] Valid Values.

### Core::X86::Apic::InterruptRequest[RequestBits] Valid Values

Bit	Description
[0]	0=Request bit not set. 1=Request bit set.
[1]	0=Request bit not set. 1=Request bit set.
[2]	0=Request bit not set. 1=Request bit set.
[3]	0=Request bit not set. 1=Request bit set.
[4]	0=Request bit not set. 1=Request bit set.
[5]	0=Request bit not set. 1=Request bit set.
[6]	0=Request bit not set. 1=Request bit set.
[7]	0=Request bit not set. 1=Request bit set.
[8]	0=Request bit not set. 1=Request bit set.
[9]	0=Request bit not set. 1=Request bit set.
[10]	0=Request bit not set. 1=Request bit set.
[11]	0=Request bit not set. 1=Request bit set.
[12]	0=Request bit not set. 1=Request bit set.
[13]	0=Request bit not set. 1=Request bit set.
[14]	0=Request bit not set. 1=Request bit set.
[15]	0=Request bit not set. 1=Request bit set.
[16]	0=Request bit not set. 1=Request bit set.
[17]	0=Request bit not set. 1=Request bit set.
[18]	0=Request bit not set. 1=Request bit set.
[19]	0=Request bit not set. 1=Request bit set.
[20]	0=Request bit not set. 1=Request bit set.
[21]	0=Request bit not set. 1=Request bit set.
[22]	0=Request bit not set. 1=Request bit set.
[23]	0=Request bit not set. 1=Request bit set.
[24]	0=Request bit not set. 1=Request bit set.
[25]	0=Request bit not set. 1=Request bit set.
[26]	0=Request bit not set. 1=Request bit set.
[27]	0=Request bit not set. 1=Request bit set.
[28]	0=Request bit not set. 1=Request bit set.
[29]	0=Request bit not set. 1=Request bit set.



[30]	0=Request bit not set. 1=Request bit set.
[31]	0=Request bit not set. 1=Request bit set.

#### APICx280 [Error Status] (Core::X86::Apic::ErrorStatus)

Writes to this register trigger an update of the register state. The value written by software is arbitrary. Each write causes the internal error state to be loaded into this register, clearing the internal error state. Consequently, a second write prior to the occurrence of another error causes the register to be overwritten with cleared data.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx280; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

Bits	Description
31:8	Reserved.
7	<b>IllegalRegAddr: illegal register address.</b> Read-write. Reset: 0. This bit indicates that an access to a nonexistent register location within this APIC was attempted. Can only be set in xAPIC mode.
6	<b>RcvdIllegalVector: received illegal vector.</b> Read-write. Reset: 0. This bit indicates that this APIC has received a message with an illegal vector (00h to 0Fh for fixed and lowest priority interrupts).
5	<b>SentIllegalVector.</b> Read-write. Reset: 0. This bit indicates that this APIC attempted to send a message with an illegal vector (00h to 0Fh for fixed and lowest priority interrupts).
4	Reserved.
3	<b>RcvAcceptError: receive accept error.</b> Read-write. Reset: 0. This bit indicates that a message received by this APIC was not accepted by this or any other APIC.
2	<b>SendAcceptError.</b> Read-write. Reset: 0. This bit indicates that a message sent by this APIC was not accepted by any APIC.
1:0	Reserved.

**APICx300 [Interrupt Command Low] (Core::X86::Apic::InterruptCommandLow)**

Reset: 0000_0000h.																			
_lthree[1:0]_core[7:0]_thread[1:0]; APICx300; APIC={Core::X86::Msr::APIC_BAR[ApicBar[47:12]] , 000h}																			
Bits	Description																		
31:20	Reserved.																		
19:18	<b>DestShrthnd: destination shorthand.</b> Read-write. Reset: 0h. <b>Description:</b> Provides a quick way to specify a destination for a message. If all including self or all excluding self is used, then destination mode is ignored and physical is automatically used. <b>ValidValues:</b>																		
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>No shorthand (Destination field).</td></tr> <tr> <td>1h</td><td>Self.</td></tr> <tr> <td>2h</td><td>All including self.</td></tr> <tr> <td>3h</td><td>All excluding self (This sends a message with a destination encoding of all 1s, so if lowest priority is used the message could end up being reflected back to this APIC).</td></tr> </table>	Value	Description	0h	No shorthand (Destination field).	1h	Self.	2h	All including self.	3h	All excluding self (This sends a message with a destination encoding of all 1s, so if lowest priority is used the message could end up being reflected back to this APIC).								
Value	Description																		
0h	No shorthand (Destination field).																		
1h	Self.																		
2h	All including self.																		
3h	All excluding self (This sends a message with a destination encoding of all 1s, so if lowest priority is used the message could end up being reflected back to this APIC).																		
17:16	<b>RemoteRdStat.</b> Read-only. Reset: 0h. <b>ValidValues:</b>																		
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Read was invalid.</td></tr> <tr> <td>1h</td><td>Delivery pending.</td></tr> <tr> <td>2h</td><td>Delivery complete and access was valid.</td></tr> <tr> <td>3h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	Read was invalid.	1h	Delivery pending.	2h	Delivery complete and access was valid.	3h	Reserved.								
Value	Description																		
0h	Read was invalid.																		
1h	Delivery pending.																		
2h	Delivery complete and access was valid.																		
3h	Reserved.																		
15	<b>TM: trigger mode.</b> Read-write. Reset: 0. 0=Edge triggered. 1=Level triggered. Indicates how this interrupt is triggered.																		
14	<b>Level.</b> Read-write. Reset: 0. 0=De-asserted. 1=Asserted.																		
13	Reserved.																		
12	<b>DS: interrupt delivery status.</b> Read-only. Reset: 0. 0=Idle. 1=Send pending. In xAPIC mode this bit is set to indicate that the interrupt has not yet been accepted by the destination core(s). Software may repeatedly write Core::X86::Apic::InterruptCommandLow without polling the DS bit; all requested IPIs are delivered.																		
11	<b>DM: destination mode.</b> Read-write. Reset: 0. 0=Physical. 1=Logical.																		
10:8	<b>MsgType.</b> Read-write. Reset: 0h. The message types are encoded as follows: <b>ValidValues:</b>																		
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Fixed.</td></tr> <tr> <td>1h</td><td>Lowest Priority.</td></tr> <tr> <td>2h</td><td>SMI.</td></tr> <tr> <td>3h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>NMI.</td></tr> <tr> <td>5h</td><td>INIT.</td></tr> <tr> <td>6h</td><td>Startup.</td></tr> <tr> <td>7h</td><td>External interrupt.</td></tr> </table>	Value	Description	0h	Fixed.	1h	Lowest Priority.	2h	SMI.	3h	Reserved.	4h	NMI.	5h	INIT.	6h	Startup.	7h	External interrupt.
Value	Description																		
0h	Fixed.																		
1h	Lowest Priority.																		
2h	SMI.																		
3h	Reserved.																		
4h	NMI.																		
5h	INIT.																		
6h	Startup.																		
7h	External interrupt.																		
7:0	<b>Vector.</b> Read-write. Reset: 00h. The vector that is sent for this interrupt source.																		

**APICx310 [Interrupt Command High] (Core::X86::Apic::InterruptCommandHigh)**

Read-write. Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx310; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

Bits	Description
31:24	<b>DestinationField.</b> Read-write. Reset: 00h. The destination encoding used when Core::X86::Apic::InterruptCommandLow[DestShrthnd] is 00b.
23:0	Reserved.

**APICx320 [LVT Timer] (Core::X86::Apic::TimerLvtEntry)**

Reset: 0001\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx320; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

Bits	Description
31:18	Reserved.
17	<b>Mode.</b> Read-write. Reset: 0. 0=One-shot. 1=Periodic.
16	<b>Mask.</b> Read-write. Reset: 1. 0=Not masked. 1=Masked.
15:13	Reserved.
12	<b>DS: interrupt delivery status.</b> Read-only, Volatile. Reset: 0. 0=Idle. 1=Send pending. (Indicates that the interrupt has not yet been accepted by the core.)
11	Reserved.
10:8	<b>MsgType: message type.</b> Read-write. Reset: 0h. See 2.1.11.2.1.14 [Generalized Local Vector Table].
7:0	<b>Vector.</b> Read-write. Reset: 00h. Interrupt vector number.

**APICx330 [LVT Thermal Sensor] (Core::X86::Apic::ThermalLvtEntry)**

Reset: 0001\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx330; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

Bits	Description
31:17	Reserved.
16	<b>Mask.</b> Read-write. Reset: 1. 0=Not masked. 1=Masked.
15:13	Reserved.
12	<b>DS: interrupt delivery status.</b> Read-only, Volatile. Reset: 0. 0=Idle. 1=Send pending. (Indicates that the interrupt has not yet been accepted by the core.)
11	Reserved.
10:8	<b>MsgType: message type.</b> Read-write. Reset: 0h. See 2.1.11.2.1.14 [Generalized Local Vector Table].
7:0	<b>Vector.</b> Read-write. Reset: 00h. Interrupt vector number.

**APICx340 [LVT Performance Monitor] (Core::X86::Apic::PerformanceCounterLvtEntry)**

Reset: 0001\_0000h.

Interrupts for this local vector table are caused by overflows of:

- Core::X86::Msr::PERF\_LEGACY\_CTL0..3(Performance Event Select [3:0]).
- Core::X86::Msr::PERF\_CTL0..5(Performance Event Select [5:0]).

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx340; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

Bits	Description
31:17	Reserved.
16	<b>Mask.</b> Read-write. Reset: 1. 0=Not masked. 1=Masked.
15:13	Reserved.
12	<b>DS: interrupt delivery status.</b> Read-only, Volatile. Reset: 0. 0=Idle. 1=Send pending. (Indicates that the interrupt has not yet been accepted by the core.)
11	Reserved.
10:8	<b>MsgType: message type.</b> Read-write. Reset: 0h. See 2.1.11.2.1.14 [Generalized Local Vector Table].
7:0	<b>Vector.</b> Read-write. Reset: 00h. Interrupt vector number.

**APICx3[5...6]0 [LVT LINT[1:0]] (Core::X86::Apic::LVTINT)**

Reset: 0001\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n0; APICx350; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n1; APICx360; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

Bits	Description
31:17	Reserved.
16	<b>Mask.</b> Read-write. Reset: 1. 0=Not masked. 1=Masked.
15	<b>TM: trigger mode.</b> Read-write. Reset: 0. 0=Edge. 1=Level.
14	<b>RmtIRR.</b> Read-only, Volatile. Reset: 0. If trigger mode is level, remote Core::X86::Apic::InterruptRequest is set when the interrupt has begun service. Remote Core::X86::Apic::InterruptRequest is cleared when the end of interrupt has occurred.
13	Reserved.
12	<b>DS: interrupt delivery status.</b> Read-only, Volatile. Reset: 0. 0=Idle. 1=Send pending. (Indicates that the interrupt has not yet been accepted by the core.)
11	Reserved.
10:8	<b>MsgType: message type.</b> Read-write. Reset: 0h. See 2.1.11.2.1.14 [Generalized Local Vector Table].
7:0	<b>Vector.</b> Read-write. Reset: 00h. Interrupt vector number.

**APICx370 [LVT Error] (Core::X86::Apic::ErrorLvtEntry)**

Reset: 0001\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx370; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

Bits	Description
31:17	Reserved.
16	<b>Mask.</b> Read-write. Reset: 1. 0=Not masked. 1=Masked.
15:13	Reserved.
12	<b>DS: interrupt delivery status.</b> Read-only, Volatile. Reset: 0. 0=Idle. 1=Send pending. (Indicates that the interrupt has not yet been accepted by the core.)
11	Reserved.
10:8	<b>MsgType: message type.</b> Read-write. Reset: 0h. See 2.1.11.2.1.14 [Generalized Local Vector Table].
7:0	<b>Vector.</b> Read-write. Reset: 00h. Interrupt vector number.

**APICx380 [Timer Initial Count] (Core::X86::Apic::TimerInitialCount)**

Read-write, Volatile. Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx380; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

Bits	Description
31:0	<b>Count.</b> Read-write, Volatile. Reset: 0000_0000h. The value copied into the current count register when the timer is loaded or reloaded.

**APICx390 [Timer Current Count] (Core::X86::Apic::TimerCurrentCount)**

Read-only, Volatile. Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx390; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

Bits	Description
31:0	<b>Count.</b> Read-only, Volatile. Reset: 0000_0000h. The current value of the counter.

**APICx3E0 [Timer Divide Configuration] (Core::X86::Apic::TimerDivideConfiguration)**

Read-write. Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx3E0; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

Bits	Description																						
31:4	Reserved.																						
3:0	<b>Div[3:0]</b> . Read-write. Reset: 0h. Div[2] is unused.																						
	<b>ValidValues:</b>																						
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Divide by 2.</td></tr> <tr> <td>1h</td><td>Divide by 4.</td></tr> <tr> <td>2h</td><td>Divide by 8.</td></tr> <tr> <td>3h</td><td>Divide by 16.</td></tr> <tr> <td>7h-4h</td><td>Reserved.</td></tr> <tr> <td>8h</td><td>Divide by 32.</td></tr> <tr> <td>9h</td><td>Divide by 64.</td></tr> <tr> <td>Ah</td><td>Divide by 128.</td></tr> <tr> <td>Bh</td><td>Divide by 1.</td></tr> <tr> <td>Fh-Ch</td><td>Reserved.</td></tr> </table>	Value	Description	0h	Divide by 2.	1h	Divide by 4.	2h	Divide by 8.	3h	Divide by 16.	7h-4h	Reserved.	8h	Divide by 32.	9h	Divide by 64.	Ah	Divide by 128.	Bh	Divide by 1.	Fh-Ch	Reserved.
Value	Description																						
0h	Divide by 2.																						
1h	Divide by 4.																						
2h	Divide by 8.																						
3h	Divide by 16.																						
7h-4h	Reserved.																						
8h	Divide by 32.																						
9h	Divide by 64.																						
Ah	Divide by 128.																						
Bh	Divide by 1.																						
Fh-Ch	Reserved.																						

**APICx400 [Extended APIC Feature] (Core::X86::Apic::ExtendedApicFeature)**

Read-only. Reset: 0004\_0007h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx400; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

Bits	Description
31:24	Reserved.
23:16	<b>ExtLvtCount: extended local vector table count</b> . Read-only. Reset: 04h. This specifies the number of extended LVT registers (Core::X86::Apic::ExtendedInterruptLvtEntries) in the local APIC.
15:3	Reserved.
2	<b>ExtApicIdCap: extended APIC ID capable</b> . Read-only. Reset: 1. 1=The processor is capable of supporting an 8-bit APIC ID, as controlled by Core::X86::Apic::ExtendedApicControl[ExtApicIdEn].
1	<b>SeoiCap: specific end of interrupt capable</b> . Read-only. Reset: 1. 1=The Core::X86::Apic::SpecificEndOfInterrupt is present.
0	<b>IerCap: interrupt enable register capable</b> . Read-only. Reset: 1. This bit indicates that the Core::X86::Apic::InterruptEnable are present. See 2.1.11.2.1.8 [Interrupt Masking].

**APICx410 [Extended APIC Control] (Core::X86::Apic::ExtendedApicControl)**

Read-write. Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx410; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]] , 000h}

Bits	Description
31:3	Reserved.
2	<b>ExtApicIdEn: extended APIC ID enable</b> . Read-write. Reset: 0. 1=Enable 8-bit APIC ID; Core::X86::Apic::ApicId[ApicId] supports an 8-bit value; an interrupt broadcast in physical destination mode requires that the IntDest[7:0] == 1111_1111b (instead of XXXX_1111b); a match in physical destination mode occurs when (IntDest[7:0] == ApicId[7:0]) instead of (IntDest[3:0] == ApicId[3:0]).
1	<b>SeoiEn</b> . Read-write. Reset: 0. 1=Enable SEOI generation when a Write to Core::X86::Apic::SpecificEndOfInterrupt is received.
0	<b>IerEn</b> . Read-write. Reset: 0. 1=Enable writes to the interrupt enable registers.

**APICx420 [Specific End Of Interrupt] (Core::X86::Apic::SpecificEndOfInterrupt)**

Read-write. Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; APICx420; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

Bits	Description
31:8	Reserved.
7:0	<b>EoiVec: end of interrupt vector.</b> Read-write. Reset: 00h. A Write to this field causes an end of interrupt cycle to be performed for the vector specified in this field. The behavior is undefined if no interrupt is pending for the specified interrupt vector.

**APICx4[8...F]0 [Interrupt Enable] (Core::X86::Apic::InterruptEnable)**

Read-write. Reset: FFFF\_FFFFh.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n0; APICx480; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n1; APICx490; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n2; APICx4A0; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n3; APICx4B0; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n4; APICx4C0; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n5; APICx4D0; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n6; APICx4E0; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n7; APICx4F0; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

Bits	Description
31:0	<b>InterruptEnableBits.</b> Read-write. Reset: FFFF_FFFFh. The interrupt enable bits can be used to enable each of the 256 interrupts.

**APICx5[0...3]0 [Extended Interrupt Local Vector Table] (Core::X86::Apic::ExtendedInterruptLvtEntries)**

Reset: 0001\_0000h.

Assignments conventions:

- APIC500 provides a local vector table entry for IBS.
- APIC510 provides a local vector table entry for error thresholding. See Core::X86::Msr::McaIntrCfg[ThresholdLvtOffset].
- APIC520 provides a local vector table entry for Deferred errors. See MCI\_CONFIG[DeferredIntType].

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n0; APICx500; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n1; APICx510; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n2; APICx520; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n3; APICx530; APIC={Core::X86::Msr::APIC\_BAR[ApicBar[47:12]], 000h}

Bits	Description
31:17	Reserved.
16	<b>Mask.</b> Read-write. Reset: 1. 0=Not masked. 1=Masked.
15:13	Reserved.
12	<b>DS: interrupt delivery status.</b> Read-only, Volatile. Reset: 0. 0=Idle. 1=Send pending. (Indicates that the interrupt has not yet been accepted by the core.)
11	Reserved.
10:8	<b>MsgType: message type.</b> Read-write. Reset: 0h. See 2.1.11.2.1.14 [Generalized Local Vector Table].
7:0	<b>Vector.</b> Read-write. Reset: 00h. Interrupt vector number.

**2.1.12 CPUID Instruction**

Processor feature capabilities and configuration information are provided through the CPUID instruction. The information is accessed by (1) selecting the CPUID function setting EAX and optionally ECX for some functions, (2) executing the CPUID instruction, and (3) reading the results in the EAX, EBX, ECX, and EDX registers. The syntax CPUID FnXXXXXXXX\_EiX[\_xYYY] refers to the function where EAX == X, and optionally ECX == Y, and the registers specified by EiX. EiX can be any single register such as {EAX, EBX, ECX, and EDX}, or a range of registers, such as E[C,B,A]X. Undefined function numbers return 0's in all 4 registers.

Unless otherwise specified, single-bit feature fields are encoded as 1=Feature is supported by the processor; 0=Feature is

not supported by the processor. CPUID functions not listed are reserved.

### 2.1.12.1 CPUID Instruction Functions

The following provides processor specific details about CPUID.

CPUID_Fn00000000_EAX [Processor Vendor and Largest Standard Function Number] (Core::X86::Cpuid::LargFuncNum)	
Read-only. Reset: Fixed,0000_0010h.	
_lthree[1:0]_core[7:0]_thread[1:0]; CPUID_Fn00000000_EAX	
Bits	Description
31:0	<b>LFuncStd: largest standard function.</b> Read-only. Reset: Fixed,0000_0010h. The largest CPUID standard function input value supported by the processor implementation.
CPUID_Fn00000000_EBX [Processor Vendor (ASCII Bytes [3:0])] (Core::X86::Cpuid::ProcVendEbx)	
Read-only. Reset: Fixed,6874_7541h.	
Core::X86::Cpuid::ProcVendEbx and Core::X86::Cpuid::ProcVendExtEbx return the same value.	
_lthree[1:0]_core[7:0]_thread[1:0]; CPUID_Fn00000000_EBX	
Bits	Description
31:0	<b>Vendor.</b> Read-only. Reset: Fixed,6874_7541h. ASCII Bytes [3:0] ("h t u A") of the string "AuthenticAMD".
CPUID_Fn00000000_ECX [Processor Vendor (ASCII Bytes [11:8])] (Core::X86::Cpuid::ProcVendEcx)	
Read-only. Reset: Fixed,444D_4163h.	
Core::X86::Cpuid::ProcVendEcx and Core::X86::Cpuid::ProcVendExtEcx return the same value.	
_lthree[1:0]_core[7:0]_thread[1:0]; CPUID_Fn00000000_ECX	
Bits	Description
31:0	<b>Vendor.</b> Read-only. Reset: Fixed,444D_4163h. ASCII Bytes [11:8] ("D M A c") of the string "AuthenticAMD".
CPUID_Fn00000000_EDX [Processor Vendor (ASCII Bytes [7:4])] (Core::X86::Cpuid::ProcVendEdx)	
Read-only. Reset: Fixed,6974_6E65h.	
Core::X86::Cpuid::ProcVendEdx and Core::X86::Cpuid::ProcVendExtEdx return the same value.	
_lthree[1:0]_core[7:0]_thread[1:0]; CPUID_Fn00000000_EDX	
Bits	Description
31:0	<b>Vendor.</b> Read-only. Reset: Fixed,6974_6E65h. ASCII Bytes [7:4] ("i t n e") of the string "AuthenticAMD".

**CPUID\_Fn00000001\_EAX [Family, Model, Stepping Identifiers] (Core::X86::Cpuid::FamModStep)**

Read-only.

Core::X86::Cpuid::FamModStep and Core::X86::Cpuid::FamModStepExt return the same value.

Family: Is an 8-bit value and is defined as: Family[7:0]={0000b,BaseFamily[3:0]}+ExtendedFamily[7:0].

- E.g., If BaseFamily[3:0] == Fh and ExtendedFamily[7:0] == 08h, then Family[7:0] = 17h.

Model: Is an 8-bit value and is defined as: Model[7:0]={ExtendedModel[3:0],BaseModel[3:0]}.

- E.g., If ExtendedModel[3:0] == 1h and BaseModel[3:0] == 8h, then Model[7:0] = 18h.
- Model numbers vary with product.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000001\_EAX

Bits	Description
31:28	Reserved.
27:20	<b>ExtFamily: extended family.</b> Read-only. Reset: 0Bh. See Family above.
19:16	<b>ExtModel: extended model.</b> Read-only. Reset: 2h. See Model above.
15:12	Reserved.
11:8	<b>BaseFamily.</b> Read-only. Reset: Fh. See Family description above.
7:4	<b>BaseModel.</b> Read-only. Reset: 4h. Model numbers vary with product.
3:0	<b>Stepping.</b> Read-only. Reset: Xh. Processor stepping (revision) for a specific model.

**CPUID\_Fn00000001\_EBX [LocalApicId, LogicalProcessorCount, CLFlush] (Core::X86::Cpuid::FeatureIdEbx)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000001\_EBX

Bits	Description
31:24	<b>LocalApicId.</b> Read-only. Reset: XXh. Initial local APIC physical ID.
23:16	<b>LogicalProcessorCount.</b> Read-only. Reset: Fixed,XXh. Specifies the number of threads in the processor as 0FFh&(Core::X86::Cpuid::SizeId[NC]+1).
15:8	<b>CLFlush.</b> Read-only. Reset: Fixed,08h. CLFLUSH size in quadwords.
7:0	Reserved.



**CPUID\_Fn00000001\_ECX [Feature Identifiers] (Core::X86::Cpuid::FeatureIdEcX)**

Read-only.

These values can be over-written by Core::X86::Msr::CPUID\_Features.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000001\_ECX

Bits	Description
31	Reserved.
30	<b>RDRAND</b> . Read-only. Reset: Fixed,1. RDRAND instruction support.
29	<b>F16C</b> . Read-only. Reset: Fixed,1. Half-precision convert instruction support.
28	<b>AVX</b> . Read-only. Reset: Fixed,1. AVX instruction support.
27	<b>OSXSAVE</b> . Read-only. Reset: X. 1=The OS has enabled support for XGETBV/XSETBV instructions to query processor extended states. OS enabled support for XGETBV/XSETBV.
26	<b>XSAVE</b> . Read-only. Reset: Fixed,1. 1=Support provided for the XSAVE, XRSTOR, XSETBV, and XGETBV instructions and the XFEATURE_ENABLED_MASK register. XSAVE (and related) instruction support.
25	<b>AES: AES instruction support</b> . Read-only. Reset: X. AES instruction support.
24	Reserved.
23	<b>POPCNT</b> . Read-only. Reset: Fixed,1. POPCNT instruction.
22	<b>MOVBE</b> . Read-only. Reset: Fixed,1. MOVBE instruction support.
21	<b>X2APIC</b> . Read-only. Reset: X. x2APIC capability.
20	<b>SSE42</b> . Read-only. Reset: Fixed,1. SSE4.2 instruction support.
19	<b>SSE41</b> . Read-only. Reset: Fixed,1. SSE4.1 instruction support.
18	Reserved.
17	<b>PCID</b> . Read-only. Reset: Fixed,0. Process context identifiers support.
16:14	Reserved.
13	<b>CMPXCHG16B</b> . Read-only. Reset: Fixed,1. CMPXCHG16B instruction.
12	<b>FMA</b> . Read-only. Reset: Fixed,1. FMA instruction support.
11:10	Reserved.
9	<b>SSSE3</b> . Read-only. Reset: Fixed,1. Supplemental SSE3 extensions.
8:4	Reserved.
3	<b>Monitor</b> . Read-only. Reset: !Core::X86::Msr::HWCR[MonMwaitDis]. Monitor/Mwait instructions.
2	Reserved.
1	<b>PCLMULQDQ</b> . Read-only. Reset: X. PCLMULQDQ instruction support.
0	<b>SSE3</b> . Read-only. Reset: Fixed,1. SSE3 extensions.

**CPUID\_Fn00000001\_EDX [Feature Identifiers] (Core::X86::Cpuid::FeatureIdEdx)**

Read-only.

These values can be over-written by Core::X86::Msr::CPUID\_Features.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000001\_EDX

Bits	Description
31:29	Reserved.
28	<b>HTT.</b> Read-only. Reset: Fixed,(Core::X86::Cpuid::SizeId[NC] != 0). 0=Single thread product (Core::X86::Cpuid::SizeId[NC] == 0). 1=Multi thread product (Core::X86::Cpuid::SizeId[NC] != 0). Hyper-threading technology.
27	Reserved.
26	<b>SSE2.</b> Read-only. Reset: Fixed,1. SSE2: SSE2 extensions.
25	<b>SSE.</b> Read-only. Reset: Fixed,1. SSE extensions.
24	<b>FXSR.</b> Read-only. Reset: Fixed,1. FXSAVE and FXRSTOR instructions.
23	<b>MMX.</b> Read-only. Reset: Fixed,1. MMX instructions
22:20	Reserved.
19	<b>CLFSH.</b> Read-only. Reset: Fixed,1. CLFLUSH instruction.
18	Reserved.
17	<b>PSE36.</b> Read-only. Reset: Fixed,1. Page-size extensions.
16	<b>PAT.</b> Read-only. Reset: Fixed,1. Page attribute table.
15	<b>CMOV.</b> Read-only. Reset: Fixed,1. Conditional move instructions, CMOV, FCOMI, FCMOV.
14	<b>MCA.</b> Read-only. Reset: Fixed,1. Machine check architecture, MCG_CAP.
13	<b>PGE.</b> Read-only. Reset: Fixed,1. Page global extension, CR4.PGE.
12	<b>MTRR.</b> Read-only. Reset: Fixed,1. Memory-type range registers.
11	<b>SysEnterSysExit.</b> Read-only. Reset: Fixed,1. SYSENTER and SYSEXIT instructions.
10	Reserved.
9	<b>APIC: advanced programmable interrupt controller (APIC) exists and is enabled.</b> Read-only. Reset: X. Core::X86::Msr::APIC_BAR[ApicEn].
8	<b>CMPXCHG8B.</b> Read-only. Reset: Fixed,1. CMPXCHG8B instruction.
7	<b>MCE.</b> Read-only. Reset: Fixed,1. Machine check exception, CR4.MCE.
6	<b>PAE.</b> Read-only. Reset: Fixed,1. Physical-address extensions (PAE).
5	<b>MSR.</b> Read-only. Reset: Fixed,1. AMD model-specific registers (MSRs), with RDMSR and WRMSR instructions.
4	<b>TSC.</b> Read-only. Reset: Fixed,1. Time Stamp Counter, RDTSC/RDTSCP instructions, CR4.TSD.
3	<b>PSE.</b> Read-only. Reset: Fixed,1. Page-size extensions (4 MB pages).
2	<b>DE.</b> Read-only. Reset: Fixed,1. Debugging extensions, IO breakpoints, CR4.DE.
1	<b>VME.</b> Read-only. Reset: Fixed,1. Virtual-mode enhancements.
0	<b>FPU.</b> Read-only. Reset: Fixed,1. x87 floating point unit on-chip.

**CPUID\_Fn00000005\_EAX [Monitor/MWait] (Core::X86::Cpuid::MonMWaitEax)**

Read-only. Reset: Fixed,0000\_0040h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000005\_EAX

Bits	Description
31:16	Reserved.
15:0	<b>MonLineSizeMin.</b> Read-only. Reset: Fixed,0040h. Smallest monitor-line size in bytes.

**CPUID\_Fn00000005\_EBX [Monitor/MWait] (Core::X86::Cpuid::MonMWaitEbx)**

Read-only. Reset: Fixed,0000\_0040h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000005\_EBX

Bits	Description
31:16	Reserved.
15:0	<b>MonLineSizeMax</b> . Read-only. Reset: Fixed,0040h. Largest monitor-line size in bytes.

**CPUID\_Fn00000005\_ECX [Monitor/MWait] (Core::X86::Cpuid::MonMWaitEcX)**

Read-only. Reset: Fixed,0000\_0003h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000005\_ECX

Bits	Description
31:2	Reserved.
1	<b>IBE</b> . Read-only. Reset: Fixed,1. Interrupt break-event.
0	<b>EMX</b> . Read-only. Reset: Fixed,1. Enumerate MONITOR/MWAIT extensions.

**CPUID\_Fn00000005\_EDX [Monitor/MWait] (Core::X86::Cpuid::MonMWaitEdx)**

Read-only. Reset: Fixed,0000\_0021h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000005\_EDX

Bits	Description
31:8	Reserved.
7:4	<b>MWaitC1SubStates</b> . Read-only. Reset: Fixed,2h. Number of C1 sub-cstates supported by MWAIT.
3:0	<b>MWaitC0SubStates</b> . Read-only. Reset: Fixed,1h. Number of C0 sub-cstates supported by MWAIT.

**CPUID\_Fn00000006\_EAX [Thermal and Power Management] (Core::X86::Cpuid::ThermalPwrMgmtEax)**

Read-only. Reset: Fixed,0000\_0004h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000006\_EAX

Bits	Description
31:3	Reserved.
2	<b>ARAT: always running APIC timer</b> . Read-only. Reset: Fixed,1. 1=Indicates support for APIC timer always running feature.
1:0	Reserved.

**CPUID\_Fn00000006\_EBX [Thermal and Power Management] (Core::X86::Cpuid::ThermalPwrMgmtEbx)**

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000006\_EBX

Bits	Description
31:0	Reserved.

**CPUID\_Fn00000006\_ECX [Thermal and Power Management] (Core::X86::Cpuid::ThermalPwrMgmtEcX)**

Read-only. Reset: Fixed,0000\_0001h.

These values can be over-written by Core::X86::MsR::CPUID\_PWR\_THERM.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000006\_ECX

Bits	Description
31:1	Reserved.
0	<b>EffFreq: effective frequency interface</b> . Read-only. Reset: Fixed,1. 1=Indicates presence of Core::X86::MsR::MPERF and Core::X86::MsR::APERF.

**CPUID\_Fn00000006\_EDX [Thermal and Power Management] (Core::X86::Cpuid::ThermalPwrMgmtEdx)**

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000006\_EDX

Bits	Description
31:0	Reserved.

**CPUID\_Fn00000007\_EAX\_x00 [Structured Extended Feature Identifiers]  
(Core::X86::Cpuid::StructExtFeatIdEax0)**

Read-only. Reset: Fixed, 0000\_0001h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000007\_EAX\_x00

Bits	Description
31:0	<b>StructExtFeatIdMax.</b> Read-only. Reset: Fixed, 0000_0001h. The largest CPUID Fn0000_0007 sub-function supported by the processor implementation.

**CPUID\_Fn00000007\_EBX\_x00 [Structured Extended Feature Identifiers]  
(Core::X86::Cpuid::StructExtFeatIdEbx0)**

Read-only. Reset: Fixed, F1BF\_97ABh.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000007\_EBX\_x00

Bits	Description
31	<b>AVX512VL.</b> Read-only. Reset: Fixed, 1. 1=Extends AVX512 instructions to 128 and 256 bits.
30	<b>AVX512BW.</b> Read-only. Reset: Fixed, 1. 1=AVX512BW packed integer instructions.
29	<b>SHA.</b> Read-only. Reset: Fixed, 1. 1=SHA Extensions available.
28	<b>AVX512CD.</b> Read-only. Reset: Fixed, 1. 1=AVX512 Conflict Detection for vectorizing loops supported.
27:25	Reserved.
24	<b>CLWB.</b> Read-only. Reset: Fixed, 1. 1=CLWB (Cache Line Write Back) instruction supported.
23	<b>CLFSHOPT.</b> Read-only. Reset: Fixed, 1. 1=CLFSHOPT (Optimized Cache Line Flush) instruction supported.
22	Reserved.
21	<b>AVX512_IFMA.</b> Read-only. Reset: Fixed, 1. 1=AVX512 integer fused mul-add instructions supported.
20	<b>SMAP.</b> Read-only. Reset: Fixed, 1. 1=Secure Mode Access Prevention is supported.
19	<b>ADX.</b> Read-only. Reset: Fixed, 1. 1=ADCX and ADOX instructions are supported.
18	<b>RDSEED.</b> Read-only. Reset: Fixed, 1. 1=RDSEED instruction is supported.
17	<b>AVX512DQ.</b> Read-only. Reset: Fixed, 1. 1=AVX512DQ packed integer instructions supported.
16	<b>AVX512F.</b> Read-only. Reset: Fixed, 1. 1=AVX512 Foundation supported.
15	<b>PQE.</b> Read-only. Reset: Fixed, 1. 1=Cache Allocation Technology is supported.
14:13	Reserved.
12	<b>PQM.</b> Read-only. Reset: Fixed, 1. 1=Platform QoS Monitoring is supported.
11	Reserved.
10	<b>INVPCID.</b> Read-only. Reset: Fixed, 1. 1=INVPCID instruction is supported.
9	<b>ERMS.</b> Read-only. Reset: Fixed, 1. 1=Enhanced REP MOVSB/STOSB is supported.
8	<b>BMI2.</b> Read-only. Reset: Fixed, 1. 1=Bit manipulation group 2 instructions are supported.
7	<b>SMEP.</b> Read-only. Reset: Fixed, 1. 1=Supervisor Mode Execution protection is supported.
6	Reserved.
5	<b>AVX2.</b> Read-only. Reset: Fixed, 1. 1=AVX extension support is supported.
4	Reserved.
3	<b>BMI1.</b> Read-only. Reset: Fixed, 1. 1=Bit manipulation group 1 instructions are supported.
2	Reserved.
1	<b>TSCADJUST.</b> Read-only. Reset: Fixed, 1. 1=Time-Stamp Counter Adjustment is supported.
0	<b>FSGSBASE.</b> Read-only. Reset: Fixed, 1. 1=FS and GS base read write instruction are supported.

**CPUID\_Fn00000007\_ECX\_x00 [Structured Extended Feature Identifier]  
(Core::X86::Cpuid::StructExtFeatIdEc0)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000007\_ECX\_x00

Bits	Description
31:29	Reserved.
28	<b>MOVDIR64B</b> . Read-only. Reset: Fixed,1. 1=MOVDIR64B instruction is supported.
27	<b>MOVDIRI</b> . Read-only. Reset: Fixed,1. 1=MOVDIRI instruction is supported.
26:25	Reserved.
24	<b>BusLock_Trap</b> . Read-only. Reset: Fixed,1. 1=Buslock Trap (generating a debug trap on bus locks that happen in CPL > 0) is supported when Core::X86::Msr::DBG_CTL_MSR[BusLockTrapEn] is set.
23	Reserved.
22	<b>RDPID</b> . Read-only. Reset: Fixed,1. 1=Read Processor ID instruction is supported.
21:15	Reserved.
14	<b>AVX512_VPOPCNTDQ</b> . Read-only. Reset: Fixed,1. 1=AVX-512 VPOPCNTD/Q instructions supported.
13	Reserved.
12	<b>AVX512_BITALG</b> . Read-only. Reset: Fixed,1. 1=AVX-512 bit algorithm instructions VPSHUFBITQMB and VPOPCNTB/W supported.
11	<b>AVX512_VNNI</b> . Read-only. Reset: Fixed,1. 1=AVX512 vector neural network instructions supported.
10	<b>VPCLMULQDQ</b> . Read-only. Reset: X. 1=Vector VPCLMULQDQ instruction supported.
9	<b>VAES</b> . Read-only. Reset: X. 1=Vector VAES(ENC DEC), VAES(ENC DEC)LAST instruction supported.
8	<b>GFNI</b> . Read-only. Reset: Fixed,1. 1=Galois Field New Instructions supported.
7	<b>CET_SS</b> . Read-only. Reset: 1. 1=Shadow stack supported.
6	<b>AVX512_VBMI2</b> . Read-only. Reset: Fixed,1. 1=AVX512 vector byte permutation instruction 2 supported.
5:4	Reserved.
3	<b>PKU</b> . Read-only. Reset: Fixed,1. 1=Protection Keys are supported.
2	<b>UMIP</b> . Read-only. Reset: Fixed,1. 1=User Mode Instruction Prevention is supported.
1	<b>AVX512_VBMI</b> . Read-only. Reset: Fixed,1. 1=AVX512 vector byte permutation instruction are supported.
0	Reserved.

**CPUID\_Fn00000007\_EDX\_x00 [Structured Extended Feature Identifiers]  
(Core::X86::Cpuid::StructExtFeatIdEdx0)**

Read-only. Reset: Fixed,1000\_0110h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000007\_EDX\_x00

Bits	Description
31:29	Reserved.
28	<b>L1D_FLUSH</b> . Read-only. Reset: Fixed,1. 1=L1D_FLUSH supported in FLUSH_CMD MSR is supported.
27:9	Reserved.
8	<b>AVX512_VP2INTERSECT</b> . Read-only. Reset: Fixed,1. 1=VP2INTERSECT instruction is supported.
7:5	Reserved.
4	<b>FSRM</b> . Read-only. Reset: Fixed,1. 1=Fast Short REP MOVSB is supported.
3:0	Reserved.

**CPUID\_Fn00000007\_EAX\_x01 [Structured Extended Feature Identifiers]  
(Core::X86::Cpuid::StructExtFeatIdEax1)**

Read-only. Reset: Fixed, 0000\_0030h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000007\_EAX\_x01

Bits	Description
31:6	Reserved.
5	<b>AVX512_BF16</b> . Read-only. Reset: Fixed, 1. 1=BFLOAT16 instructions are supported.
4	<b>AVXVNNI</b> . Read-only. Reset: Fixed, 1. 1=AVX vector neural network instructions are supported.
3:0	Reserved.

**CPUID\_Fn00000007\_EBX\_x01 [Structured Extended Feature Identifiers]  
(Core::X86::Cpuid::StructExtFeatIdEbx1)**

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000007\_EBX\_x01

Bits	Description
31:0	Reserved.

**CPUID\_Fn00000007\_ECX\_x01 [Structured Extended Feature Identifier]  
(Core::X86::Cpuid::StructExtFeatIdEcX1)**

Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000007\_ECX\_x01

Bits	Description
31:0	Reserved.

**CPUID\_Fn00000007\_EDX\_x01 [Structured Extended Feature Identifiers]  
(Core::X86::Cpuid::StructExtFeatIdEdx1)**

Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000007\_EDX\_x01

Bits	Description
31:0	Reserved.

**CPUID\_Fn0000000B\_EAX\_x00 [Extended Topology Enumeration] (Core::X86::Cpuid::ExtTopEnumEax0)**

Read-only.

CPUID Fn0000\_000B\_E[D,C,B,A]X\_x[2:0] specifies the hierarchy of logical cores from the SMT level through the processor socket level.

Software determines the presence of CPUID Fn0000\_000B if (CPUID Fn0000\_000B\_EBX\_x0[31:0] != 0). Software reads CPUID Fn0000\_000B\_E[C,B,A]X for ascending values of ECX until (CPUID Fn0000\_000B\_EBX[LogProcAtThisLevel] == 0).

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000B\_EAX\_x00

Bits	Description
31:5	Reserved.
4:0	<b>CoreMaskWidth</b> . Read-only. Number of bits to shift ExtendedApicId right to get unique topology ID of the next instance of the current level type. Reset: SMT ? 01h : 00h.

**CPUID\_Fn0000000B\_EBX\_x00 [Extended Topology Enumeration] (Core::X86::Cpuid::ExtTopEnumEbx0)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000B\_EBX\_x00

Bits	Description
31:16	Reserved.
15:0	<b>LogProcAtThisLevel</b> . Read-only. Number of threads in a core. Reset: SMT ? 2 : 0001h.

**CPUID\_Fn0000000B\_ECX\_x00 [Extended Topology Enumeration] (Core::X86::Cpuid::ExtTopEnumEcx0)**

Read-only. Reset: Fixed,0000\_0100h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000B\_ECX\_x00

Bits	Description										
31:16	Reserved.										
15:8	<b>LevelType.</b> Read-only. Reset: Fixed,01h. <b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00h</td><td>Invalid</td></tr> <tr> <td>01h</td><td>Thread</td></tr> <tr> <td>02h</td><td>Processor</td></tr> <tr> <td>FFh-03h</td><td>Reserved.</td></tr> </table>	Value	Description	00h	Invalid	01h	Thread	02h	Processor	FFh-03h	Reserved.
Value	Description										
00h	Invalid										
01h	Thread										
02h	Processor										
FFh-03h	Reserved.										
7:0	<b>EcxFVal.</b> Read-only. Reset: Fixed,00h. ECX input value.										

**CPUID\_Fn0000000B\_EAX\_x01 [Extended Topology Enumeration] (Core::X86::Cpuid::ExtTopEnumEax1)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000B\_EAX\_x01

Bits	Description
31:5	Reserved.
4:0	<b>CoreMaskWidth.</b> Read-only. Reset: XXXXXb. ExtendedApicId right shift value.

**CPUID\_Fn0000000B\_EBX\_x01 [Extended Topology Enumeration] (Core::X86::Cpuid::ExtTopEnumEbx1)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000B\_EBX\_x01

Bits	Description
31:16	Reserved.
15:0	<b>LogProcAtThisLevel.</b> Read-only. Reset: XXXXh. Number of logical cores in processor socket.

**CPUID\_Fn0000000B\_ECX\_x01 [Extended Topology Enumeration] (Core::X86::Cpuid::ExtTopEnumEcx1)**

Read-only. Reset: Fixed,0000\_0201h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000B\_ECX\_x01

Bits	Description										
31:16	Reserved.										
15:8	<b>LevelType.</b> Read-only. Reset: Fixed,02h. <b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00h</td><td>Invalid</td></tr> <tr> <td>01h</td><td>Thread</td></tr> <tr> <td>02h</td><td>Processor</td></tr> <tr> <td>FFh-03h</td><td>Reserved.</td></tr> </table>	Value	Description	00h	Invalid	01h	Thread	02h	Processor	FFh-03h	Reserved.
Value	Description										
00h	Invalid										
01h	Thread										
02h	Processor										
FFh-03h	Reserved.										
7:0	<b>EcxFVal.</b> Read-only. Reset: Fixed,01h. ECX input value.										

**CPUID\_Fn0000000B\_EAX\_x02 [Extended Topology Enumeration] (Core::X86::Cpuid::ExtTopEnumEax2)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000B\_EAX\_x02

Bits	Description
31:5	Reserved.
4:0	<b>CoreMaskWidth.</b> Read-only. Reset: Fixed,00h. Zero indicates no more levels.



**CPUID\_Fn0000000B\_EBX\_x02 [Extended Topology Enumeration] (Core::X86::Cpuid::ExtTopEnumEbx2)**

Read-only. Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000B\_EBX\_x02

Bits	Description
31:16	Reserved.
15:0	<b>LogProcAtThisLevel.</b> Read-only. Reset: 0000h. Zero indicates no more levels.

**CPUID\_Fn0000000B\_ECX\_x02 [Extended Topology Enumeration] (Core::X86::Cpuid::ExtTopEnumEcxc2)**

Read-only. Reset: Fixed,0000\_0002h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000B\_ECX\_x02

Bits	Description										
31:16	Reserved.										
15:8	<b>LevelType.</b> Read-only. Reset: Fixed,00h. <b>ValidValues:</b> <table data-bbox="181 655 1518 877"> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00h</td><td>Invalid</td></tr> <tr> <td>01h</td><td>Thread</td></tr> <tr> <td>02h</td><td>Processor</td></tr> <tr> <td>FFh-03h</td><td>Reserved.</td></tr> </table>	Value	Description	00h	Invalid	01h	Thread	02h	Processor	FFh-03h	Reserved.
Value	Description										
00h	Invalid										
01h	Thread										
02h	Processor										
FFh-03h	Reserved.										
7:0	<b>EcxcVal.</b> Read-only. Reset: Fixed,02h. ECX input value.										

**CPUID\_Fn0000000B\_EDX [Extended Topology Enumeration] (Core::X86::Cpuid::ExtTopEnumEdx)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000B\_EDX

Bits	Description
31:0	<b>ExtendedLocalApicId: extended APIC ID.</b> Read-only. Reset: XXXX_XXXXh. Extended APIC_ID.

**CPUID\_Fn0000000D\_EAX\_x00 [Processor Extended State Enumeration] (Core::X86::Cpuid::ProcExtStateEnumEax00)**

Read-only. Reset: Fixed,0000\_02E7h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EAX\_x00

Bits	Description																																	
31:0	<b>XFeatureSupportedMask[31:0]</b> . Read-only. Reset: Fixed,0000_02E7h. Each set bit indicates the corresponding bit in register XCR0[31:0] is settable.																																	
	<b>ValidValues:</b>																																	
	<table><tr><th>Bit</th><th>Name</th><th>Description</th></tr><tr><td>[0]</td><td>X87</td><td>X87 Support.</td></tr><tr><td>[1]</td><td>SSE</td><td>128-bit SSE Support.</td></tr><tr><td>[2]</td><td>AVX</td><td>256-bit AVX support.</td></tr><tr><td>[4:3]</td><td></td><td>Reserved.</td></tr><tr><td>[5]</td><td>KREGS</td><td>KREGS</td></tr><tr><td>[6]</td><td>ZMMHI</td><td>ZMMHI</td></tr><tr><td>[7]</td><td>HIZMM</td><td>HIZMM</td></tr><tr><td>[8]</td><td></td><td>Reserved.</td></tr><tr><td>[9]</td><td>MPK</td><td>Memory Protection Keys. See Core::X86::Cpuid::StructExtFeatIdEcxc0[PKU] for the availability of MPK feature support.</td></tr><tr><td>[31:10]</td><td></td><td>Reserved.</td></tr></table>	Bit	Name	Description	[0]	X87	X87 Support.	[1]	SSE	128-bit SSE Support.	[2]	AVX	256-bit AVX support.	[4:3]		Reserved.	[5]	KREGS	KREGS	[6]	ZMMHI	ZMMHI	[7]	HIZMM	HIZMM	[8]		Reserved.	[9]	MPK	Memory Protection Keys. See Core::X86::Cpuid::StructExtFeatIdEcxc0[PKU] for the availability of MPK feature support.	[31:10]		Reserved.
	Bit	Name	Description																															
	[0]	X87	X87 Support.																															
	[1]	SSE	128-bit SSE Support.																															
	[2]	AVX	256-bit AVX support.																															
	[4:3]		Reserved.																															
	[5]	KREGS	KREGS																															
	[6]	ZMMHI	ZMMHI																															
	[7]	HIZMM	HIZMM																															
	[8]		Reserved.																															
[9]	MPK	Memory Protection Keys. See Core::X86::Cpuid::StructExtFeatIdEcxc0[PKU] for the availability of MPK feature support.																																
[31:10]		Reserved.																																



**CPUID\_Fn0000000D\_EBX\_x00 [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEbx00)**

Read-only, Volatile.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EBX\_x00

Bits	Description
31:0	<b>XFeatureEnabledSizeMax.</b> Read-only, Volatile. Reset: XXXX_XXXXh. <b>Description:</b> Size in bytes of an uncompact XSAVE/XRSTOR area for all features enabled in the XCR0 register. IF (XCR0[MPK] == 1) return EBX=0000_0988h ELSIF (XCR0[HIZMM] == 1) return EBX=0000_0980h ELSIF (XCR0[ZMMHI] == 1) return EBX=0000_0580h ELSIF (XCR0[KREGS] == 1) return EBX=0000_0380h ELSIF (XCR0[AVX] == 1) return EBX=0000_0340h ELSIF (XCR0[SSE] == 1) or (XCR0[X87] == 1) return EBX=0000_0240h // legacy header + X87/SSE size END

**CPUID\_Fn0000000D\_ECX\_x00 [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEcx00)**

Read-only. Reset: Fixed, 0000\_0988h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_ECX\_x00

Bits	Description
31:0	<b>XFeatureSupportedSizeMax.</b> Read-only. Reset: Fixed, 0000_0988h. Size of legacy header + X87/SSE + AVX + KREGS + ZMMHI + HIZMM + MPK.

**CPUID\_Fn0000000D\_EDX\_x00 [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEdx00)**

Read-only. Reset: Fixed, 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EDX\_x00

Bits	Description
31:0	<b>XFeatureSupportedMask[63:32].</b> Read-only. Reset: Fixed, 0000_0000h. Each set bit indicates the corresponding bit in register XCR0[63:32] is settable.

**CPUID\_Fn0000000D\_EAX\_x01 [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEax01)**

Read-only. Reset: Fixed, 0000\_000Fh.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EAX\_x01

Bits	Description
31:4	Reserved.
3	<b>XSAVES.</b> Read-only. Reset: Fixed, 1. XSAVES, XRSTORS, and XSS supported.
2	<b>XGETBV.</b> Read-only. Reset: Fixed, 1. XGETBV with ECX=1 supported.
1	<b>XSAVEC.</b> Read-only. Reset: Fixed, 1. XSAVEC and compact XRSTOR supported.
0	<b>XSAVEOPT.</b> Read-only. Reset: Fixed, 1. XSAVEOPT is available.

**CPUID\_Fn0000000D\_EBX\_x01 [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEbx01)**

Read-only, Volatile.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EBX\_x01

Bits	Description
31:0	<b>XFeatureEnabledSizeMax.</b> Read-only, Volatile. Reset: XXXX_XXXXh. <b>Description:</b> Reset is EBX = 0000_0240h + ((XCR0[AVX] == 1) ? 0000_0100h : 0) + ((XCR0[KREGS] == 1) ? 0000_0040h : 0) + ((XCR0[ZMMHI] == 1) ? 0000_0200h : 0) + ((XCR0[HIZMM] == 1) ? 0000_0400h : 0) + ((XCR0[MPK] == 1) ? 0000_0008h : 0) + ((XSS[CET_U] == 1) ? 0000_0010h : 0) + ((XSS[CET_S] == 1) ? 0000_0018h : 0).

**CPUID\_Fn0000000D\_ECX\_x01 [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEcX01)**

Read-only. Reset: Fixed, 0000\_1800h.

Each set bit indicates the corresponding bit in register XSS[31:0] is settable.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_ECX\_x01

Bits	Description										
31:0	<b>MaskXss.</b> Read-only. Reset: Fixed, 0000_1800h. Mask[31:0] of settable XSS bits. <b>ValidValues:</b> <table border="1"> <thead> <tr> <th>Bit</th><th>Description</th></tr> </thead> <tbody> <tr> <td>[10:0]</td><td>Reserved.</td></tr> <tr> <td>[11]</td><td>CET for user mode.</td></tr> <tr> <td>[12]</td><td>CET for supervisor mode.</td></tr> <tr> <td>[31:13]</td><td>Reserved.</td></tr> </tbody> </table>	Bit	Description	[10:0]	Reserved.	[11]	CET for user mode.	[12]	CET for supervisor mode.	[31:13]	Reserved.
Bit	Description										
[10:0]	Reserved.										
[11]	CET for user mode.										
[12]	CET for supervisor mode.										
[31:13]	Reserved.										

**CPUID\_Fn0000000D\_EDX\_x01 [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEdx01)**

Read-only. Reset: Fixed, 0000\_0000h.

Each set bit indicates the corresponding bit in register XSS[63:32] is settable.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EDX\_x01

Bits	Description
31:0	<b>MaskXss.</b> Read-only. Reset: Fixed, 0000_0000h. Mask[63:32] of settable XSS bits.

**CPUID\_Fn0000000D\_EAX\_x02 [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEax02)**

Read-only. Reset: Fixed, 0000\_0100h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EAX\_x02

Bits	Description
31:0	<b>YmmHiSaveStateOffset.</b> Read-only. Reset: Fixed, 0000_0100h. YMM[31:16] save state byte size.

**CPUID\_Fn0000000D\_EBX\_x02 [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEbx02)**

Read-only. Reset: Fixed, 0000\_0240h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EBX\_x02

Bits	Description
31:0	<b>YmmHiSaveStateOffset.</b> Read-only. Reset: Fixed, 0000_0240h. YMM[31:16] save state uncompact byte offset.

**CPUID\_Fn0000000D\_ECX\_x02 [Processor Extended State Enumeration]**  
**(Core::X86::Cpuid::ProcExtStateEnumEcxE02)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_ECX\_x02

Bits	Description
31:2	Reserved.
1	<b>YmmHiAligned.</b> Read-only. Reset: Fixed,0. 0=YMM_hi state (YMM[31:16]) is not automatically aligned to a 64-byte boundary on compacted saves/restores. 1=YMM_hi state (YMM[31:16]) is automatically aligned to a 64-byte boundary on compacted saves/restores.
0	<b>XStateSupervisor.</b> Read-only. Reset: Fixed,0. 1=This xstate is Supervisor State.

**CPUID\_Fn0000000D\_EDX\_x02 [Processor Extended State Enumeration]**  
**(Core::X86::Cpuid::ProcExtStateEnumEdxE02)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EDX\_x02

Bits	Description
31:0	Reserved.

**CPUID\_Fn0000000D\_EAX\_x05 [Processor Extended State Enumeration]**  
**(Core::X86::Cpuid::ProcExtStateEnumEaxE05)**

Read-only. Reset: Fixed,0000\_0040h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EAX\_x05

Bits	Description
31:0	<b>KREGSSaveStateSize.</b> Read-only. Reset: Fixed,0000_0040h. KREGS save state byte size.

**CPUID\_Fn0000000D\_EBX\_x05 [Processor Extended State Enumeration]**  
**(Core::X86::Cpuid::ProcExtStateEnumEbxE05)**

Read-only. Reset: Fixed,0000\_0340h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EBX\_x05

Bits	Description
31:0	<b>KREGSSaveStateOffset.</b> Read-only. Reset: Fixed,0000_0340h. KREGS save state uncompact byte offset.

**CPUID\_Fn0000000D\_ECX\_x05 [Processor Extended State Enumeration]**  
**(Core::X86::Cpuid::ProcExtStateEnumEcxE05)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_ECX\_x05

Bits	Description
31:2	Reserved.
1	<b>KREGSAligned.</b> Read-only. Reset: Fixed,0. 0=KREGS state is not automatically aligned to a 64-byte boundary on compacted saves/restores. 1=KREGS state is automatically aligned to a 64-byte boundary on compacted saves/restores.
0	<b>XStateSupervisor.</b> Read-only. Reset: Fixed,0. 1=This xstate is Supervisor State.

**CPUID\_Fn0000000D\_EDX\_x05 [Processor Extended State Enumeration]**  
**(Core::X86::Cpuid::ProcExtStateEnumEdxE05)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EDX\_x05

Bits	Description
31:0	Reserved.

**CPUID\_Fn0000000D\_EAX\_x06 [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEax06)**

Read-only. Reset: Fixed,0000\_0200h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EAX\_x06

Bits	Description
31:0	<b>ZMMHISaveStateSize.</b> Read-only. Reset: Fixed,0000_0200h. ZMMHI save state byte size.

**CPUID\_Fn0000000D\_EBX\_x06 [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEbx06)**

Read-only. Reset: Fixed,0000\_0380h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EBX\_x06

Bits	Description
31:0	<b>ZMMHISaveStateOffset.</b> Read-only. Reset: Fixed,0000_0380h. ZMMHI save state uncompact byte offset.

**CPUID\_Fn0000000D\_ECX\_x06 [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEcX06)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_ECX\_x06

Bits	Description
31:2	Reserved.
1	<b>ZMMHISAligned.</b> Read-only. Reset: Fixed,0. 0=ZMMHI state is not automatically aligned to a 64-byte boundary on compacted saves/restores. 1=ZMMHI is automatically aligned to a 64-byte boundary on compacted saves/restores.
0	<b>XStateSupervisor.</b> Read-only. Reset: Fixed,0. 1=This xstate is Supervisor State.

**CPUID\_Fn0000000D\_EDX\_x06 [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEdx06)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EDX\_x06

Bits	Description
31:0	Reserved.

**CPUID\_Fn0000000D\_EAX\_x07 [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEax07)**

Read-only. Reset: Fixed,0000\_0400h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EAX\_x07

Bits	Description
31:0	<b>HIZMMSaveStateSize.</b> Read-only. Reset: Fixed,0000_0400h. HIZMM save state byte size.

**CPUID\_Fn0000000D\_EBX\_x07 [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEbx07)**

Read-only. Reset: Fixed,0000\_0580h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EBX\_x07

Bits	Description
31:0	<b>HIZMMSaveStateOffset.</b> Read-only. Reset: Fixed,0000_0580h. HIZMM save state uncompact byte offset.

**CPUID\_Fn0000000D\_ECX\_x07 [Processor Extended State Enumeration]**  
**(Core::X86::Cpuid::ProcExtStateEnumEcX07)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_ECX\_x07

Bits	Description
31:2	Reserved.
1	<b>HIZMMAligned.</b> Read-only. Reset: Fixed,0. 0=HIZMM state is not automatically aligned to a 64-byte boundary on compacted saves/restores. 1=HIZMM state is automatically aligned to a 64-byte boundary on compacted saves/restores.
0	<b>XStateSupervisor.</b> Read-only. Reset: Fixed,0. 1=This xstate is Supervisor State.

**CPUID\_Fn0000000D\_EDX\_x07 [Processor Extended State Enumeration]**  
**(Core::X86::Cpuid::ProcExtStateEnumEdX07)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EDX\_x07

Bits	Description
31:0	Reserved.

**CPUID\_Fn0000000D\_EAX\_x09 [Processor Extended State Enumeration]**  
**(Core::X86::Cpuid::ProcExtStateEnumEax09)**

Read-only. Reset: Fixed,0000\_0008h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EAX\_x09

Bits	Description
31:0	<b>MpkSaveStateSize.</b> Read-only. Reset: Fixed,0000_0008h. MPK save state byte size.

**CPUID\_Fn0000000D\_EBX\_x09 [Processor Extended State Enumeration]**  
**(Core::X86::Cpuid::ProcExtStateEnumEbx09)**

Read-only. Reset: Fixed,0000\_0980h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EBX\_x09

Bits	Description
31:0	<b>MpKSaveStateOffset.</b> Read-only. Reset: Fixed,0000_0980h. MPK save state uncompact byte offset.

**CPUID\_Fn0000000D\_ECX\_x09 [Processor Extended State Enumeration]**  
**(Core::X86::Cpuid::ProcExtStateEnumEcX09)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_ECX\_x09

Bits	Description
31:2	Reserved.
1	<b>XState64ByteAligned.</b> Read-only. Reset: Fixed,0. 1=This xstate will always be 64-byte aligned in compacted memops.
0	<b>XStateSupervisor.</b> Read-only. Reset: Fixed,0. 1=This xstate is Supervisor State.

**CPUID\_Fn0000000D\_EDX\_x09 [Processor Extended State Enumeration]**  
**(Core::X86::Cpuid::ProcExtStateEnumEdX09)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EDX\_x09

Bits	Description
31:0	Reserved.

**CPUID\_Fn0000000D\_EAX\_x0B [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEax0B)**

Read-only. Reset: Fixed,0000\_0010h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EAX\_x0B

Bits	Description
------	-------------

31:0	<b>CetUserSize.</b> Read-only. Reset: Fixed,0000_0010h. : CET user size.
------	--

**CPUID\_Fn0000000D\_EBX\_x0B [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEbx0B)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EBX\_x0B

Bits	Description
------	-------------

31:0	<b>CetUserOffset.</b> Read-only. Reset: Fixed,0000_0000h. CET user byte offset.
------	---

**CPUID\_Fn0000000D\_ECX\_x0B [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEcxB)**

Read-only. Reset: Fixed,0000\_0001h.

Processor Extended State Enumeration for CET\_U.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_ECX\_x0B

Bits	Description
------	-------------

31:2	Reserved.
------	-----------

1	<b>XState64ByteAligned.</b> Read-only. Reset: Fixed,0. 1=This xstate will always be 64-byte aligned in compacted memops.
---	--

0	<b>XStateSupervisor.</b> Read-only. Reset: Fixed,1. 1=This xstate is Supervisor State.
---	--

**CPUID\_Fn0000000D\_EDX\_x0B [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEdxB)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EDX\_x0B

Bits	Description
------	-------------

31:0	Reserved.
------	-----------

**CPUID\_Fn0000000D\_EAX\_x0C [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEax0C)**

Read-only. Reset: Fixed,0000\_0018h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EAX\_x0C

Bits	Description
------	-------------

31:0	<b>CetSprvrSize.</b> Read-only. Reset: Fixed,0000_0018h. CET supervisor size.
------	---

**CPUID\_Fn0000000D\_EBX\_x0C [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEbx0C)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EBX\_x0C

Bits	Description
------	-------------

31:0	<b>CetSprvrOffset.</b> Read-only. Reset: Fixed,0000_0000h. CET supervisor byte offset.
------	--

**CPUID\_Fn0000000D\_ECX\_x0C [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEcxC)**

Read-only. Reset: Fixed,0000\_0001h.

Processor Extended State Enumeration for CET\_S.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_ECX\_x0C

Bits	Description
31:2	Reserved.
1	<b>XState64ByteAligned</b> . Read-only. Reset: Fixed,0. 1=This xstate will always be 64-byte aligned in compacted memops.
0	<b>XStateSupervisor</b> . Read-only. Reset: Fixed,1. 1=This xstate is Supervisor State.

**CPUID\_Fn0000000D\_EDX\_x0C [Processor Extended State Enumeration]  
(Core::X86::Cpuid::ProcExtStateEnumEdxC)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000D\_EDX\_x0C

Bits	Description
31:0	Reserved.

**CPUID\_Fn0000000F\_EAX\_x00 [Resource Director Technology Monitor Capability]  
(Core::X86::Cpuid::RsrcDirTechMonCapEax0)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000F\_EAX\_x00

Bits	Description
31:0	Reserved.

**CPUID\_Fn0000000F\_EBX\_x00 [Resource Director Technology Monitor Capability]  
(Core::X86::Cpuid::RsrcDirTechMonCapEbx0)**

Read-only. Reset: Fixed,0000\_0FFFh.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000F\_EBX\_x00

Bits	Description
31:0	<b>RmidMaxRange</b> . Read-only. Reset: Fixed,0000_0FFFh. RMID maximum within this processor for all types.

**CPUID\_Fn0000000F\_ECX\_x00 [Resource Director Technology Monitor Capability]  
(Core::X86::Cpuid::RsrcDirTechMonCapEcxC)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000F\_ECX\_x00

Bits	Description
31:0	Reserved.

**CPUID\_Fn0000000F\_EDX\_x00 [Resource Director Technology Monitor Capability]  
(Core::X86::Cpuid::RsrcDirTechMonCapEdxC)**

Read-only. Reset: Fixed,0000\_0002h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000F\_EDX\_x00

Bits	Description
31:2	Reserved.
1	<b>L3CacheRDT</b> . Read-only. Reset: Fixed,1. L3 Cache RDT Monitoring.
0	Reserved.



**CPUID\_Fn0000000F\_EAX\_x01 [Resource Director Technology L3 Monitor Capability]  
(Core::X86::Cpuid::RsrcDirTechMonCapEax1)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000F\_EAX\_x01

Bits	Description
31:9	Reserved.
8	<b>OverflowBit.</b> Read-only. Reset: Fixed,0. 1=Indicates Core::X86::Msr::QM_CTR bit[61] is an overflow bit.
7:0	<b>CounterSize.</b> Read-only. Reset: Fixed,20. Encode counter width offset from bit 24.
<b>ValidValues:</b>	
<b>Value</b>	<b>Description</b>
00h	Family/Model/Stepping should be used to determine counter size.
26h-01h	<Value>+24-bit counters.
FFh-27h	Reserved.

**CPUID\_Fn0000000F\_EBX\_x01 [Resource Director Technology L3 Monitor Capability]  
(Core::X86::Cpuid::RsrcDirTechMonCapEbx1)**

Read-only. Reset: Fixed,0000\_0040h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000F\_EBX\_x01

Bits	Description
31:0	<b>ConverFactor.</b> Read-only. Reset: Fixed,0000_0040h. Conversion Factor.

**CPUID\_Fn0000000F\_ECX\_x01 [Resource Director Technology L3 Monitor Capability]  
(Core::X86::Cpuid::RsrcDirTechMonCapEcxc1)**

Read-only. Reset: Fixed,0000\_0FFFh.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000F\_ECX\_x01

Bits	Description
31:0	<b>RmidMaxRange.</b> Read-only. Reset: Fixed,0000_0FFFh. RMID Maximum Range of this resource.

**CPUID\_Fn0000000F\_EDX\_x01 [Resource Director Technology L3 Monitor Capability]  
(Core::X86::Cpuid::RsrcDirTechMonCapEdxc1)**

Read-only. Reset: Fixed,0000\_0007h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn0000000F\_EDX\_x01

Bits	Description
31:3	Reserved.
2	<b>L3CacheLocalBndwdthMon.</b> Read-only. Reset: Fixed,1. L3 Local Bandwidth monitoring.
1	<b>L3CacheTotalBndwdthMon.</b> Read-only. Reset: Fixed,1. L3 Total Bandwidth monitoring.
0	<b>L3CacheOccpncyMon.</b> Read-only. Reset: Fixed,1. L3 occupancy monitoring.

**CPUID\_Fn00000010\_EAX\_x00 [Resource Director Technology Allocation Enumeration]  
(Core::X86::Cpuid::RsrcDirTechAllocEnumEax0)**

Read-only. Reset: Fixed,0000\_0000h.

Software determines the presence of CPUID Fn0000\_0010 if (CPUID Fn0000\_0010\_EBX\_x0[31:0] != 0). Software reads CPUID Fn0000\_0010\_E[D,C,B,A]X for ascending values of ECX until (CPUID Fn0000\_0010\_EBX[LogProcAtThisLevel] == 0).

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000010\_EAX\_x00

Bits	Description
31:0	Reserved.



**CPUID\_Fn00000010\_EBX\_x00 [Resource Director Technology Allocation Enumeration]  
(Core::X86::Cpuid::RsrcDirTechAllocEnumEbx0)**

Read-only. Reset: 0000\_0002h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000010\_EBX\_x00

Bits	Description
31:3	Reserved.
2	<b>L2CacheAllocTech.</b> Read-only. Reset: 0. L2 Cache Allocation Technology.
1	<b>L3CacheAllocTech.</b> Read-only. Reset: 1. L3 Cache Allocation Technology.
0	Reserved.

**CPUID\_Fn00000010\_ECX\_x00 [Resource Director Technology Allocation Enumeration]  
(Core::X86::Cpuid::RsrcDirTechAllocEnumEcx0)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000010\_ECX\_x00

Bits	Description
31:0	Reserved.

**CPUID\_Fn00000010\_EDX\_x00 [Resource Director Technology Allocation Enumeration]  
(Core::X86::Cpuid::RsrcDirTechAllocEnumEdx0)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000010\_EDX\_x00

Bits	Description
31:0	Reserved.

**CPUID\_Fn00000010\_EAX\_x01 [Resource Director Technology L3 Allocation Enumeration]  
(Core::X86::Cpuid::RsrcDirTechAllocEnumEax1)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000010\_EAX\_x01

Bits	Description
31:5	Reserved.
4:0	<b>CapacityMask.</b> Read-only. Reset: Fixed,0Fh. Capacity bitmask length.

**CPUID\_Fn00000010\_EBX\_x01 [Resource Director Technology L3 Allocation Enumeration]  
(Core::X86::Cpuid::RsrcDirTechAllocEnumEbx1)**

Read-only. Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000010\_EBX\_x01

Bits	Description
31:0	<b>AllocUnits.</b> Read-only. Reset: 0000_0000h. Allocation Units.

**CPUID\_Fn00000010\_ECX\_x01 [Resource Director Technology L3 Allocation Enumeration]  
(Core::X86::Cpuid::RsrcDirTechAllocEnumEcx1)**

Read-only. Reset: Fixed,0000\_0004h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000010\_ECX\_x01

Bits	Description
31:3	Reserved.
2	<b>CDP.</b> Read-only. Reset: Fixed,1. Code and data prioritization.
1:0	Reserved.

**CPUID\_Fn00000010\_EDX\_x01 [Resource Director Technology L3 Allocation Enumeration] (Core::X86::Cpuid::RsrcDirTechAllocEnumEdx1)**

Read-only. Reset: Fixed,0000\_000Fh.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn00000010\_EDX\_x01

Bits	Description
31:16	Reserved.
15:0	<b>HCS.</b> Read-only. Reset: Fixed,000Fh. Highest Class Of Service (COS) supported.

**CPUID\_Fn80000000\_EAX [Largest Extended Function Number] (Core::X86::Cpuid::LargExtFuncNum)**

Read-only. Reset: Fixed,8000\_0028h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000000\_EAX

Bits	Description
31:0	<b>LFuncExt: largest extended function.</b> Read-only. Reset: Fixed,8000_0028h. The largest CPUID extended function input value supported by the processor implementation.

**CPUID\_Fn80000000\_EBX [Processor Vendor (ASCII Bytes [3:0])] (Core::X86::Cpuid::ProcVendExtEbx)**

Read-only. Reset: Fixed,6874\_7541h.

Core::X86::Cpuid::ProcVendEbx and Core::X86::Cpuid::ProcVendExtEbx return the same value.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000000\_EBX

Bits	Description
31:0	<b>Vendor.</b> Read-only. Reset: Fixed,6874_7541h. ASCII Bytes [3:0] ("h t u A") of the string "AuthenticAMD".

**CPUID\_Fn80000000\_ECX [Processor Vendor (ASCII Bytes [11:8])] (Core::X86::Cpuid::ProcVendExtEcx)**

Read-only. Reset: Fixed,444D\_4163h.

Core::X86::Cpuid::ProcVendEcx and Core::X86::Cpuid::ProcVendExtEcx return the same value.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000000\_ECX

Bits	Description
31:0	<b>Vendor.</b> Read-only. Reset: Fixed,444D_4163h. ASCII Bytes [11:8] ("D M A c") of the string "AuthenticAMD".

**CPUID\_Fn80000000\_EDX [Processor Vendor (ASCII Bytes [7:4])] (Core::X86::Cpuid::ProcVendExtEdx)**

Read-only. Reset: Fixed,6974\_6E65h.

Core::X86::Cpuid::ProcVendEdx and Core::X86::Cpuid::ProcVendExtEdx return the same value.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000000\_EDX

Bits	Description
31:0	<b>Vendor.</b> Read-only. Reset: Fixed,6974_6E65h. ASCII Bytes [7:4] ("i t n e") of the string "AuthenticAMD".

**CPUID\_Fn80000001\_EAX [Family, Model, Stepping Identifiers] (Core::X86::Cpuid::FamModStepExt)**

Read-only.

Core::X86::Cpuid::FamModStep and Core::X86::Cpuid::FamModStepExt return the same value. See Core::X86::Cpuid::FamModStep.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000001\_EAX

Bits	Description
31:28	Reserved.
27:20	<b>ExtFamily: extended family.</b> Read-only. Reset: 0Bh. See Core::X86::Cpuid::FamModStep description of Family.
19:16	<b>ExtModel: extended model.</b> Read-only. Reset: 2h. See Core::X86::Cpuid::FamModStep description of ExtModel.
15:12	Reserved.
11:8	<b>BaseFamily.</b> Read-only. Reset: Fh. See Core::X86::Cpuid::FamModStep description of Family.
7:4	<b>BaseModel.</b> Read-only. Reset: Xh. Model numbers vary with product.
3:0	<b>Stepping.</b> Read-only. Reset: Xh. Processor stepping (revision) for a specific model.

**CPUID\_Fn80000001\_EBX [BrandId Identifier] (Core::X86::Cpuid::BrandId)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000001\_EBX

Bits	Description								
31:28	<b>PkgType: package type.</b> Read-only. Reset: Xh. Specifies the package type. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>3h-0h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>SP5</td></tr> <tr> <td>Fh-5h</td><td>Reserved.</td></tr> </table>	Value	Description	3h-0h	Reserved.	4h	SP5	Fh-5h	Reserved.
Value	Description								
3h-0h	Reserved.								
4h	SP5								
Fh-5h	Reserved.								
27:0	Reserved.								

CPUID_Fn80000001_ECX [Feature Identifiers] (Core::X86::Cpuid::FeatureExtIdEcX)	
Read-only.	
These values can be over-written by Core::X86::Msr::CPUID_ExtFeatures.	
_lthree[1:0]_core[7:0]_thread[1:0]; CPUID_Fn80000001_ECX	
Bits	Description
31	Reserved.
30	<b>AdMskExtn: address mask extension support for instruction breakpoint.</b> Read-only. Reset: Fixed,1. Indicates support for address mask extension (to 32 bits and to all 4 DRs) for instruction breakpoints.
29	<b>MwaitExtended.</b> Read-only. Reset: !Core::X86::Msr::HWCR[MonMwaitDis]. 1=MWAITX and MONITORX capability is supported.
28	<b>PerfCtrExtLLC: Last Level Cache performance counter extensions.</b> Read-only. Reset: Fixed,1. 1=Indicates support for L3 performance counter extensions.
27	<b>PerfTsc.</b> Read-only. Reset: Fixed,0. Performance time-stamp counter supported.
26	<b>DataBreakpointExtension.</b> Read-only. Reset: Fixed,1. 1=Indicates data breakpoint support for Core::X86::Msr::DR0_ADDR_MASK, Core::X86::Msr::DR1_ADDR_MASK, Core::X86::Msr::DR2_ADDR_MASK and Core::X86::Msr::DR3_ADDR_MASK.
25	Reserved.
24	<b>PerfCtrExtDF: data fabric performance counter extensions support.</b> Read-only. Reset: Fixed,1. 1=Indicates support for Core::X86::Msr::DF_PERF_CTL and Core::X86::Msr::DF_PERF_CTR.
23	<b>PerfCtrExtCore: core performance counter extensions support.</b> Read-only. Reset: Fixed,1. 1=Indicates support for Core::X86::Msr::PERF_CTL0 - 5 and Core::X86::Msr::PERF_CTR.
22	<b>TopologyExtensions: topology extensions support.</b> Read-only. Reset: Fixed,1. 1=Indicates support for Core::X86::Cpuid::CachePropEax0 and Core::X86::Cpuid::ExtApicId.
21:18	Reserved.
17	<b>TCE.</b> Read-only. Reset: Fixed,1. Translation cache extension.
16	<b>FMA4.</b> Read-only. Reset: Fixed,0. Four-operand FMA instruction support.
15	<b>LWP.</b> Read-only. Reset: Fixed,0. Lightweight profiling support.
14	Reserved.
13	<b>WDT.</b> Read-only. Reset: Fixed,1. Watchdog timer support.
12	<b>SKINIT.</b> Read-only. Reset: Fixed,1. SKINIT and STGI support.
11	<b>XOP.</b> Read-only. Reset: Fixed,0. Extended operation support.
10	<b>IBS.</b> Read-only. Reset: Fixed,1. Instruction Based Sampling.
9	<b>OSVW.</b> Read-only. Reset: Fixed,1. OS Visible Work-around support.
8	<b>ThreeDNowPrefetch.</b> Read-only. Reset: Fixed,1. Prefetch and PrefetchW instructions.
7	<b>MisAlignSse.</b> Read-only. Reset: Fixed,1. Misaligned SSE Mode.
6	<b>SSE4A.</b> Read-only. Reset: Fixed,1. EXTRQ, INSERTQ, MOVNTSS, and MOVNTSD instruction support.
5	<b>ABM: advanced bit manipulation.</b> Read-only. Reset: Fixed,1. LZCNT instruction support.
4	<b>AltMovCr8.</b> Read-only. Reset: Fixed,1. LOCK MOV CR0 means MOV CR8.
3	<b>ExtApicSpace.</b> Read-only. Reset: Fixed,1. Extended APIC register space.
2	<b>SVM: Secure Virtual Mode feature.</b> Read-only. Reset: Fixed,1. Indicates support for: VMRUN, VMLOAD, VMSAVE, CLGI, VMMCALL, and INVLPGA.
1	<b>CmpLegacy.</b> Read-only. Reset: Fixed,(Core::X86::Cpuid::SizeId[NC] > 0). 0=Single core product (Core::X86::Cpuid::SizeId[NC] == 0). 1=Multi core product (Core::X86::Cpuid::SizeId[NC] != 0). Core multi-processing legacy mode.
0	<b>LahfSahf.</b> Read-only. Reset: Fixed,1. LAHF and SAHF instruction support in 64-bit mode.

**CPUID\_Fn80000001\_EDX [Feature Identifiers] (Core::X86::Cpuid::FeatureExtIdEdx)**

Read-only.

These values can be over-written by Core::X86::Msr::CPUID\_ExtFeatures.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000001\_EDX

Bits	Description
31	<b>ThreeDNow</b> . Read-only. Reset: Fixed,0. 3DNow! instructions.
30	<b>ThreeDNowExt</b> . Read-only. Reset: Fixed,0. AMD extensions to 3DNow! instructions.
29	<b>LM</b> . Read-only. Reset: Fixed,1. Long Mode.
28	Reserved.
27	<b>RDTSCP</b> . Read-only. Reset: Fixed,1. RDTSCP instruction.
26	<b>Page1GB</b> . Read-only. Reset: Fixed,1. 1-GB large page support.
25	<b>FFXSR</b> . Read-only. Reset: Fixed,1. FXSAVE and FXRSTOR instruction optimizations.
24	<b>FXSR</b> . Read-only. Reset: Fixed,1. FXSAVE and FXRSTOR instructions.
23	<b>MMX</b> . Read-only. Reset: Fixed,1. MMX instructions.
22	<b>MmxExt</b> . Read-only. Reset: Fixed,1. AMD extensions to MMX instructions.
21	Reserved.
20	<b>NX</b> . Read-only. Reset: Fixed,1. No-execute page protection.
19:18	Reserved.
17	<b>PSE36</b> . Read-only. Reset: Fixed,1. Page-size extensions.
16	<b>PAT</b> . Read-only. Reset: Fixed,1. Page attribute table.
15	<b>CMOV</b> . Read-only. Reset: Fixed,1. Conditional move instructions, CMOV, FCOMI, FCMOV.
14	<b>MCA</b> . Read-only. Reset: Fixed,1. Machine check architecture, MCG_CAP.
13	<b>PGE</b> . Read-only. Reset: Fixed,1. Page global extension, CR4.PGE.
12	<b>MTRR</b> . Read-only. Reset: Fixed,1. Memory-type range registers.
11	<b>SysCallSysRet</b> . Read-only. Reset: Fixed,1. SYSCALL and SYSRET instructions.
10	Reserved.
9	<b>APIC: advanced programmable interrupt controller (APIC) exists and is enabled</b> . Read-only. Reset: X. Reset is Core::X86::Msr::APIC_BAR[ApicEn].
8	<b>CMPXCHG8B</b> . Read-only. Reset: Fixed,1. CMPXCHG8B instruction.
7	<b>MCE</b> . Read-only. Reset: Fixed,1. Machine Check Exception, CR4.MCE.
6	<b>PAE</b> . Read-only. Reset: Fixed,1. Physical-address extensions (PAE).
5	<b>MSR</b> . Read-only. Reset: Fixed,1. Model-specific registers (MSRs), with RDMSR and WRMSR instructions.
4	<b>TSC</b> . Read-only. Reset: Fixed,1. Time stamp counter, RDTSC/RDTSCP instructions, CR4.TSD.
3	<b>PSE</b> . Read-only. Reset: Fixed,1. Page-size extensions (4 MB pages).
2	<b>DE</b> . Read-only. Reset: Fixed,1. Debugging extensions, IO breakpoints, CR4.DE.
1	<b>VME</b> . Read-only. Reset: Fixed,1. Virtual-mode enhancements.
0	<b>FPU</b> . Read-only. Reset: Fixed,1. x87 floating point unit on-chip.

**CPUID\_Fn80000002\_EAX [Processor Name String Identifier (Bytes [3:0])]  
(Core::X86::Cpuid::ProcNameStr0Eax)**

Read-only.

Is an alias of Core::X86::Msr::ProcNameString\_n0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000002\_EAX

Bits	Description
31:24	<b>ProcNameByte3.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n0[CpuNameString3]. Processor name, byte3.
23:16	<b>ProcNameByte2.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n0[CpuNameString2]. Processor name, byte2.
15:8	<b>ProcNameByte1.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n0[CpuNameString1]. Processor name, byte1.
7:0	<b>ProcNameByte0.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n0[CpuNameString0]. Processor name, byte0.

**CPUID\_Fn80000002\_EBX [Processor Name String Identifier (Bytes [7:4])]  
(Core::X86::Cpuid::ProcNameStr0Ebx)**

Read-only.

Is an alias of Core::X86::Msr::ProcNameString\_n0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000002\_EBX

Bits	Description
31:24	<b>ProcNameByte7.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n0[CpuNameString7]. Processor name, byte 7.
23:16	<b>ProcNameByte6.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n0[CpuNameString6]. Processor name, byte 6.
15:8	<b>ProcNameByte5.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n0[CpuNameString5]. Processor name, byte 5.
7:0	<b>ProcNameByte4.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n0[CpuNameString4]. Processor name, byte 4.

**CPUID\_Fn80000002\_ECX [Processor Name String Identifier (Bytes [11:8])]  
(Core::X86::Cpuid::ProcNameStr0Ecx)**

Read-only.

Is an alias of Core::X86::Msr::ProcNameString\_n1.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000002\_ECX

Bits	Description
31:24	<b>ProcNameByte11.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n1[CpuNameString3]. Processor name, byte 11.
23:16	<b>ProcNameByte10.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n1[CpuNameString2]. Processor name, byte 10.
15:8	<b>ProcNameByte9.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n1[CpuNameString1]. Processor name, byte 9.
7:0	<b>ProcNameByte8.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n1[CpuNameString0]. Processor name, byte 8.

**CPUID\_Fn80000002\_EDX [Processor Name String Identifier (Bytes [15:12])]  
(Core::X86::Cpuid::ProcNameStr0Edx)**

Read-only.

Is an alias of Core::X86::Msr::ProcNameString\_n1.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000002\_EDX

Bits	Description
31:24	<b>ProcNameByte15.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n1[CpuNameString7]. Processor name, byte 15.
23:16	<b>ProcNameByte14.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n1[CpuNameString6]. Processor name, byte 14.
15:8	<b>ProcNameByte13.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n1[CpuNameString5]. Processor name, byte 13.
7:0	<b>ProcNameByte12.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n1[CpuNameString4]. Processor name, byte 12.

**CPUID\_Fn80000003\_EAX [Processor Name String Identifier (Bytes [19:16])]  
(Core::X86::Cpuid::ProcNameStr1Eax)**

Read-only.

Is an alias of Core::X86::Msr::ProcNameString\_n2.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000003\_EAX

Bits	Description
31:24	<b>ProcNameByte19.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n2[CpuNameString3]. Processor name, byte 19.
23:16	<b>ProcNameByte18.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n2[CpuNameString2]. Processor name, byte 18.
15:8	<b>ProcNameByte17.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n2[CpuNameString1]. Processor name, byte 17.
7:0	<b>ProcNameByte16.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n2[CpuNameString0]. Processor name, byte 16.

**CPUID\_Fn80000003\_EBX [Processor Name String Identifier (Bytes [23:20])]  
(Core::X86::Cpuid::ProcNameStr1Ebx)**

Read-only.

Is an alias of Core::X86::Msr::ProcNameString\_n2.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000003\_EBX

Bits	Description
31:24	<b>ProcNameByte23.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n2[CpuNameString7]. Processor name, byte 23.
23:16	<b>ProcNameByte22.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n2[CpuNameString6]. Processor name, byte 22.
15:8	<b>ProcNameByte21.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n2[CpuNameString5]. Processor name, byte 21.
7:0	<b>ProcNameByte20.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n2[CpuNameString4]. Processor name, byte 20.



**CPUID\_Fn80000003\_ECX [Processor Name String Identifier (Bytes [27:24])]  
(Core::X86::Cpuid::ProcNameStr1EcX)**

Read-only.

Is an alias of Core::X86::Msr::ProcNameString\_n3.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000003\_ECX

Bits	Description
31:24	<b>ProcNameByte27.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n3[CpuNameString3]. Processor name, byte 27.
23:16	<b>ProcNameByte26.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n3[CpuNameString2]. Processor name, byte 26.
15:8	<b>ProcNameByte25.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n3[CpuNameString1]. Processor name, byte 25.
7:0	<b>ProcNameByte24.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n3[CpuNameString0]. Processor name, byte 24.

**CPUID\_Fn80000003\_EDX [Processor Name String Identifier (Bytes [31:28])]  
(Core::X86::Cpuid::ProcNameStr1EdX)**

Read-only.

Is an alias of Core::X86::Msr::ProcNameString\_n3.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000003\_EDX

Bits	Description
31:24	<b>ProcNameByte31.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n3[CpuNameString7]. Processor name, byte 31.
23:16	<b>ProcNameByte30.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n3[CpuNameString6]. Processor name, byte 30.
15:8	<b>ProcNameByte29.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n3[CpuNameString5]. Processor name, byte 29.
7:0	<b>ProcNameByte28.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n3[CpuNameString4]. Processor name, byte 28.

**CPUID\_Fn80000004\_EAX [Processor Name String Identifier (Bytes [35:32])]  
(Core::X86::Cpuid::ProcNameStr2Eax)**

Read-only.

Is an alias of Core::X86::Msr::ProcNameString\_n4.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000004\_EAX

Bits	Description
31:24	<b>ProcNameByte35.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n4[CpuNameString3]. Processor name, byte 35.
23:16	<b>ProcNameByte34.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n4[CpuNameString2]. Processor name, byte 34.
15:8	<b>ProcNameByte33.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n4[CpuNameString1]. Processor name, byte 33.
7:0	<b>ProcNameByte32.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n4[CpuNameString0]. Processor name, byte 32.



**CPUID\_Fn80000004\_EBX [Processor Name String Identifier (Bytes [39:36])]  
(Core::X86::Cpuid::ProcNameStr2Ebx)**

Read-only.

Is an alias of Core::X86::Msr::ProcNameString\_n4.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000004\_EBX

Bits	Description
31:24	<b>ProcNameByte39.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n4[CpuNameString7]. Processor name, byte 39.
23:16	<b>ProcNameByte38.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n4[CpuNameString6]. Processor name, byte 38.
15:8	<b>ProcNameByte37.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n4[CpuNameString5]. Processor name, byte 37.
7:0	<b>ProcNameByte36.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n4[CpuNameString4]. Processor name, byte 36.

**CPUID\_Fn80000004\_ECX [Processor Name String Identifier (Bytes [43:40])]  
(Core::X86::Cpuid::ProcNameStr2EcX)**

Read-only.

Is an alias of Core::X86::Msr::ProcNameString\_n5.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000004\_ECX

Bits	Description
31:24	<b>ProcNameByte43.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n5[CpuNameString3]. Processor name, byte 43.
23:16	<b>ProcNameByte42.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n5[CpuNameString2]. Processor name, byte 42.
15:8	<b>ProcNameByte41.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n5[CpuNameString1]. Processor name, byte 41.
7:0	<b>ProcNameByte40.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n5[CpuNameString0]. Processor name, byte 40.

**CPUID\_Fn80000004\_EDX [Processor Name String Identifier (Bytes [47:44])]  
(Core::X86::Cpuid::ProcNameStr2Edx)**

Read-only.

Is an alias of Core::X86::Msr::ProcNameString\_n5.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000004\_EDX

Bits	Description
31:24	<b>ProcNameByte47.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n5[CpuNameString7]. Processor name, byte 47.
23:16	<b>ProcNameByte46.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n5[CpuNameString6]. Processor name, byte 46.
15:8	<b>ProcNameByte45.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n5[CpuNameString5]. Processor name, byte 45.
7:0	<b>ProcNameByte44.</b> Read-only. Reset: Core::X86::Msr::ProcNameString_n5[CpuNameString4]. Processor name, byte 44.

**CPUID\_Fn80000005\_EAX [L1 TLB 2M/4M Identifiers] (Core::X86::Cpuid::L1Tlb2M4M)**

Read-only.

This function provides the processor's first level cache and TLB characteristics for each core.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000005\_EAX

Bits	Description
31:24	<b>L1DTlb2and4MAssoc: data TLB associativity for 2 MB and 4 MB pages.</b> Read-only. Reset: Fixed,FFh. See Core::X86::Cpuid::L1DcId[L1DcAssoc].
23:16	<b>L1DTlb2and4MSize: data TLB number of entries for 2 MB and 4 MB pages.</b> Read-only. Reset: Fixed,96. The value returned is for the number of entries available for the 2 MB page size; 4 MB pages require two 2 MB entries, so the number of entries available for the 4 MB page size is one-half the returned value.
15:8	<b>L1ITlb2and4MAssoc: instruction TLB associativity for 2 MB and 4 MB pages.</b> Read-only. Reset: Fixed,FFh. See Core::X86::Cpuid::L1DcId[L1DcAssoc].
7:0	<b>L1ITlb2and4MSize: instruction TLB number of entries for 2 MB and 4 MB pages.</b> Read-only. Reset: Fixed,64. The value returned is for the number of entries available for the 2 MB page size; 4 MB pages require two 2 MB entries, so the number of entries available for the 4 MB page size is one-half the returned value.

**CPUID\_Fn80000005\_EBX [L1 TLB 4K Identifiers] (Core::X86::Cpuid::L1Tlb4K)**

Read-only.

See Core::X86::Cpuid::L1Tlb2M4M.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000005\_EBX

Bits	Description
31:24	<b>L1DTlb4KAssoc.</b> Read-only. Reset: Fixed,FFh. Data TLB associativity for 4 KB pages. See Core::X86::Cpuid::L1DcId[L1DcAssoc].
23:16	<b>L1DTlb4KSize.</b> Read-only. Reset: Fixed,96. Data TLB number of entries for 4 KB pages.
15:8	<b>L1ITlb4KAssoc.</b> Read-only. Reset: Fixed,FFh. Instruction TLB associativity for 4 KB pages. See Core::X86::Cpuid::L1DcId[L1DcAssoc].
7:0	<b>L1ITlb4KSize.</b> Read-only. Reset: Fixed,64. Instruction TLB number of entries for 4 KB pages.

**CPUID\_Fn80000005\_ECX [L1 Data Cache Identifiers] (Core::X86::Cpuid::L1DcId)**

Read-only.

This function provides first level cache characteristics for each core.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000005\_ECX

Bits	Description														
31:24	<b>L1DcSize.</b> Read-only. Reset: Fixed,30h. L1 data cache size in KB.														
23:16	<b>L1DcAssoc.</b> Read-only. Reset: Fixed,0Ch. L1 data cache associativity. <b>ValidValues:</b>														
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00h</td><td>Reserved.</td></tr> <tr> <td>01h</td><td>1 way (direct mapped)</td></tr> <tr> <td>02h</td><td>2 way</td></tr> <tr> <td>03h</td><td>3 way</td></tr> <tr> <td>FEh-04h</td><td>&lt;Value&gt; way</td></tr> <tr> <td>FFh</td><td>Fully associative</td></tr> </table>	Value	Description	00h	Reserved.	01h	1 way (direct mapped)	02h	2 way	03h	3 way	FEh-04h	<Value> way	FFh	Fully associative
Value	Description														
00h	Reserved.														
01h	1 way (direct mapped)														
02h	2 way														
03h	3 way														
FEh-04h	<Value> way														
FFh	Fully associative														
15:8	<b>L1DcLinesPerTag.</b> Read-only. Reset: Fixed,01h. L1 data cache lines per tag.														
7:0	<b>L1DcLineSize.</b> Read-only. Reset: Fixed,64. L1 data cache line size in bytes.														

**CPUID\_Fn80000005\_EDX [L1 Instruction Cache Identifiers] (Core::X86::Cpuid::L1CId)**

Read-only.

This function provides first level cache characteristics for each core.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000005\_EDX

Bits	Description																
31:24	<b>L1IcSize.</b> Read-only. Reset: Fixed,32. L1 instruction cache size KB.																
23:16	<b>L1IcAssoc.</b> Read-only. Reset: Fixed,8. L1 instruction cache associativity.																
	<b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00h</td><td>Reserved.</td></tr> <tr> <td>01h</td><td>1 way (direct mapped)</td></tr> <tr> <td>02h</td><td>2 way</td></tr> <tr> <td>03h</td><td>3 way</td></tr> <tr> <td>04h</td><td>4 way</td></tr> <tr> <td>FEh-05h</td><td>&lt;Value&gt; way</td></tr> <tr> <td>FFh</td><td>Fully associative</td></tr> </table>	Value	Description	00h	Reserved.	01h	1 way (direct mapped)	02h	2 way	03h	3 way	04h	4 way	FEh-05h	<Value> way	FFh	Fully associative
Value	Description																
00h	Reserved.																
01h	1 way (direct mapped)																
02h	2 way																
03h	3 way																
04h	4 way																
FEh-05h	<Value> way																
FFh	Fully associative																
15:8	<b>L1IcLinesPerTag.</b> Read-only. Reset: Fixed,01h. L1 instruction cache lines per tag.																
7:0	<b>L1IcLineSize.</b> Read-only. Reset: Fixed,64. L1 instruction cache line size in bytes.																

**CPUID\_Fn80000006\_EAX [L2 TLB 2M/4M Identifiers] (Core::X86::Cpuid::L2Tlb2M4M)**

Read-only.

This function provides the processor's second level cache and TLB characteristics for each core.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000006\_EAX

Bits	Description								
31:28	<b>L2DTlb2and4MAssoc: L2 data TLB associativity for 2 MB and 4 MB pages.</b> Read-only. Reset: Fixed,4h.								
	<b>ValidValues:</b>								
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>3h-0h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>4 – 5 ways</td></tr> <tr> <td>Fh-5h</td><td>Reserved.</td></tr> </table>	Value	Description	3h-0h	Reserved.	4h	4 – 5 ways	Fh-5h	Reserved.
Value	Description								
3h-0h	Reserved.								
4h	4 – 5 ways								
Fh-5h	Reserved.								
27:16	<b>L2DTlb2and4MSize: L2 data TLB number of entries for 2 MB and 4 MB pages.</b> Read-only. Reset: Fixed,080h. The number of entries available for the 2 MB page size is 32 times the returned value; 4 MB pages require two 2 MB entries, so the number of entries available for the 4 MB page size is one-half the number of 2 MB entries.								
15:12	<b>L2ITlb2and4MAssoc: L2 instruction TLB associativity for 2 MB and 4 MB pages.</b> Read-only. Reset: Fixed,2.								
	<b>ValidValues:</b>								
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>1h-0h</td><td>Reserved.</td></tr> <tr> <td>2h</td><td>2 ways</td></tr> <tr> <td>Fh-3h</td><td>Reserved.</td></tr> </table>	Value	Description	1h-0h	Reserved.	2h	2 ways	Fh-3h	Reserved.
Value	Description								
1h-0h	Reserved.								
2h	2 ways								
Fh-3h	Reserved.								
11:0	<b>L2ITlb2and4MSize: L2 instruction TLB number of entries for 2 MB and 4 MB pages.</b> Read-only. Reset: Fixed,040h. The number of entries available for the 2 MB page size is 32 times the returned value; 4 MB pages require two 2 MB entries, so the number of entries available for the 4 MB page size is one-half the number of 2 MB entries.								

**CPUID\_Fn80000006\_EBX [L2 TLB 4K Identifiers] (Core::X86::Cpuid::L2Tlb4K)**

Read-only.

This function provides the processor's second level cache and TLB characteristics for each core.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000006\_EBX

Bits	Description								
31:28	<b>L2DTlb4KAssoc.</b> Read-only. Reset: Fixed,6h. L2 data TLB associativity for 4 KB pages. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>5h-0h</td><td>Reserved.</td></tr> <tr> <td>6h</td><td>8 – 15 ways</td></tr> <tr> <td>Fh-7h</td><td>Reserved.</td></tr> </table>	Value	Description	5h-0h	Reserved.	6h	8 – 15 ways	Fh-7h	Reserved.
Value	Description								
5h-0h	Reserved.								
6h	8 – 15 ways								
Fh-7h	Reserved.								
27:16	<b>L2DTlb4KSize.</b> Read-only. Reset: Fixed,128. The L2 data TLB number of entries for 4 KB pages is 32 times the returned value.								
15:12	<b>L2ITlb4KAssoc.</b> Read-only. Reset: Fixed,4. L2 instruction TLB associativity for 4 KB pages. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>3h-0h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>4 – 5 ways</td></tr> <tr> <td>Fh-5h</td><td>Reserved.</td></tr> </table>	Value	Description	3h-0h	Reserved.	4h	4 – 5 ways	Fh-5h	Reserved.
Value	Description								
3h-0h	Reserved.								
4h	4 – 5 ways								
Fh-5h	Reserved.								
11:0	<b>L2ITlb4KSize.</b> Read-only. Reset: Fixed,64. The L2 instruction TLB number of entries for 4 KB pages is 32 times the returned value.								

**CPUID\_Fn80000006\_ECX [L2 Cache Identifiers] (Core::X86::Cpuid::L2CacheId)**

Read-only.

This function provides second level cache characteristics for each core.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000006\_ECX

Bits	Description																				
31:16	<b>L2Size.</b> Read-only. Reset: Fixed,0400h. L2 cache size in KB. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00FFh-0000h</td><td>Reserved.</td></tr> <tr> <td>0100h</td><td>256 KB</td></tr> <tr> <td>01FFh-0101h</td><td>Reserved.</td></tr> <tr> <td>0200h</td><td>512 KB</td></tr> <tr> <td>03FFh-0201h</td><td>Reserved.</td></tr> <tr> <td>0400h</td><td>1 MB</td></tr> <tr> <td>07FFh-0401h</td><td>Reserved.</td></tr> <tr> <td>0800h</td><td>2 MB</td></tr> <tr> <td>FFFFh-0801h</td><td>Reserved.</td></tr> </table>	Value	Description	00FFh-0000h	Reserved.	0100h	256 KB	01FFh-0101h	Reserved.	0200h	512 KB	03FFh-0201h	Reserved.	0400h	1 MB	07FFh-0401h	Reserved.	0800h	2 MB	FFFFh-0801h	Reserved.
Value	Description																				
00FFh-0000h	Reserved.																				
0100h	256 KB																				
01FFh-0101h	Reserved.																				
0200h	512 KB																				
03FFh-0201h	Reserved.																				
0400h	1 MB																				
07FFh-0401h	Reserved.																				
0800h	2 MB																				
FFFFh-0801h	Reserved.																				
15:12	<b>L2Assoc.</b> Read-only. Reset: Fixed,8. L2 cache associativity. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>7h-0h</td><td>Reserved.</td></tr> <tr> <td>8h</td><td>16 – 31 ways</td></tr> <tr> <td>Fh-9h</td><td>Reserved.</td></tr> </table>	Value	Description	7h-0h	Reserved.	8h	16 – 31 ways	Fh-9h	Reserved.												
Value	Description																				
7h-0h	Reserved.																				
8h	16 – 31 ways																				
Fh-9h	Reserved.																				
11:8	<b>L2LinesPerTag.</b> Read-only. Reset: Fixed,1h. L2 cache lines per tag.																				
7:0	<b>L2LineSize.</b> Read-only. Reset: Fixed,64. L2 cache line size in bytes.																				

**CPUID\_Fn80000006\_EDX [L3 Cache Identifiers] (Core::X86::Cpuid::L3CacheId)**

Read-only.

This function provides third level cache characteristics shared by all cores of a processor.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000006\_EDX

Bits	Description								
31:18	<b>L3Size: L3 cache size.</b> Read-only. Reset: XXXXh. The L3 cache size in 512 KB units. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0000h</td><td>Disabled.</td></tr> <tr> <td>3FFFh-0001h</td><td>(&lt;Value&gt; * 0.5) MB</td></tr> </table>	Value	Description	0000h	Disabled.	3FFFh-0001h	(<Value> * 0.5) MB		
Value	Description								
0000h	Disabled.								
3FFFh-0001h	(<Value> * 0.5) MB								
17:16	Reserved.								
15:12	<b>L3Assoc.</b> Read-only. Reset: Fixed,9h. There are insufficient available encodings to represent all possible L3 associativities. Please refer to Core::X86::Cpuid::CachePropEbx3[CacheNumWays]. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>8h-0h</td><td>Reserved.</td></tr> <tr> <td>9h</td><td>Invalid, not reported here.</td></tr> <tr> <td>Fh-Ah</td><td>Reserved.</td></tr> </table>	Value	Description	8h-0h	Reserved.	9h	Invalid, not reported here.	Fh-Ah	Reserved.
Value	Description								
8h-0h	Reserved.								
9h	Invalid, not reported here.								
Fh-Ah	Reserved.								
11:8	<b>L3LinesPerTag.</b> Read-only. Reset: Fixed,1h. L3 cache lines per tag.								
7:0	<b>L3LineSize.</b> Read-only. Reset: Fixed,64. L3 cache line size in bytes.								

**CPUID\_Fn80000007\_EAX [Reserved] (Core::X86::Cpuid::ProcFeedbackCap)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000007\_EAX

Bits	Description
31:0	Reserved.

**CPUID\_Fn80000007\_EBX [RAS Capabilities] (Core::X86::Cpuid::RasCap)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000007\_EBX

Bits	Description
31:4	Reserved.
3	<b>ScalableMca.</b> Read-only. Reset: Fixed,1. 0=Scalable MCA is not supported. 1=Scalable MCA is supported.
2	<b>HWA.</b> Read-only. Reset: Fixed,0. Hardware assert supported.
1	<b>SUCCOR: Software uncorrectable error containment and recovery capability.</b> Read-only. Reset: X. The processor supports software containment of uncorrectable errors through context synchronizing data poisoning and deferred error interrupts; MSR Core::X86::Msr::McaIntrCfg, MCA_STATUS[Deferred] and MCA_STATUS[Poison] exist.
0	<b>McaOverflowRecov: MCA overflow recovery support.</b> Read-only. Reset: Fixed,1. 0=MCA overflow conditions require software to shutdown the system. 1=MCA overflow conditions (MCA_STATUS[Overflow] == 1) are not fatal; software may safely ignore such conditions.

**CPUID\_Fn80000007\_ECX [Advanced Power Management Information] (Core::X86::Cpuid::ApmInfoEcX)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000007\_ECX

Bits	Description
31:0	<b>CpuPwrSampleTimeRatio.</b> Read-only. Reset: Fixed,0000_0000h. Specifies the ratio of the compute unit power accumulator sample period to the TSC counter period.

**CPUID\_Fn80000007\_EDX [Advanced Power Management Information] (Core::X86::Cpuid::ApmInfoEdx)**

Read-only.

This function provides advanced power management feature identifiers.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000007\_EDX

Bits	Description
31:16	Reserved.
15	<b>FastCPPC</b> . Read-only. Reset: Fixed,1. When set, indicates that Fast CPPC is supported.
14	<b>RAPL</b> . Read-only. Reset: Fixed,1. Running average power limit.
13	<b>ConnectedStandby</b> . Read-only. Reset: Fixed,1. Connected Standby.
12	<b>ProcPowerReporting</b> . Read-only. Reset: Fixed,0. Core power reporting interface supported.
11	<b>ProcFeedbackInterface: processor feedback interface</b> . Read-only. Reset: Fixed,0. 1=Indicates support for processor feedback interface; Core::X86::Cpuid::ProcFeedbackCap.
10	<b>EffFreqRO: read-only effective frequency interface</b> . Read-only. Reset: Fixed,1. Indicates presence of Core::X86::Msr::MPerfReadOnly and Core::X86::Msr::APerfReadOnly.
9	<b>CPB: core performance boost</b> . Read-only. Reset: X. 1=Indicates presence of Core::X86::Msr::HWCR[CpbDis] and support for core performance boost.
8	<b>TscInvariant: TSC invariant</b> . Read-only. Reset: Fixed,1. The TSC rate is invariant.
7	<b>HwPstate: hardware P-state control</b> . Read-only. Reset: Fixed,1. Core::X86::Msr::PStateCurLim, Core::X86::Msr::PStateCtl and Core::X86::Msr::PStateStat exist.
6	<b>OneHundredMHzSteps</b> . Read-only. Reset: Fixed,0. 100 MHz multiplier Control.
5	Reserved.
4	<b>TM</b> . Read-only. Reset: Fixed,1. Hardware thermal control (HTC)
3	<b>TTP</b> . Read-only. Reset: Fixed,1. THERMTRIP.
2:1	Reserved.
0	<b>TS</b> . Read-only. Reset: Fixed,1. Temperature sensor.

**CPUID\_Fn80000008\_EAX [Long Mode Address Size Identifiers] (Core::X86::Cpuid::LongModeInfo)**

Read-only. Reset: Fixed,0000\_3030h.

This provides information about the maximum physical and linear address width supported by the processor.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000008\_EAX

Bits	Description						
31:24	Reserved.						
23:16	<b>GuestPhysAddrSize</b> . Read-only. Reset: Fixed,00h. Maximum guest physical byte address size in bits. <b>ValidValues:</b>						
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00h</td><td>The maximum guest physical address size defined by PhysAddrSize.</td></tr> <tr> <td>FFh-01h</td><td>The maximum guest physical address size defined by GuestPhysAddrSize.</td></tr> </table>	Value	Description	00h	The maximum guest physical address size defined by PhysAddrSize.	FFh-01h	The maximum guest physical address size defined by GuestPhysAddrSize.
Value	Description						
00h	The maximum guest physical address size defined by PhysAddrSize.						
FFh-01h	The maximum guest physical address size defined by GuestPhysAddrSize.						
15:8	<b>LinAddrSize</b> . Read-only. Reset: Fixed,30h. Maximum linear byte address size in bits.						
7:0	<b>PhysAddrSize</b> . Read-only. Reset: Fixed,30h. Maximum physical byte address size in bits.						

**CPUID\_Fn80000008\_EBX [Extended Feature Extensions ID EBX] (Core::X86::Cpuid::FeatureExtIdEbx)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000008\_EBX

Bits	Description
31	<b>BranchSample.</b> Read-only. Reset: Fixed,0. 1=Branch sampling feature supported. Branch sampling has been replaced with LBRv2.
30	<b>IBPB_RET.</b> Read-only. Reset: Fixed,1. When set, the processor clears the return address predictor on Core::X86::Msr::PRED_CMD[IBPB].
29	<b>BTC_NO.</b> Read-only. Reset: Fixed,1. Processor is not vulnerable to Branch Type Confusion.
28	<b>PSFD.</b> Read-only. Reset: Fixed,1. Predictive Store Forward Disable. See Core::X86::Msr::SPEC_CTRL[PSFD].
27	<b>CPPC.</b> Read-only. Reset: 1. Collaborative Processor Performance Control.
26:25	Reserved.
24	<b>SSBD: Speculative Store Bypass Disable.</b> Read-only. Reset: Fixed,1.
23	<b>PPIN: PPIN support.</b> Read-only. Reset: X. 0=PPIN capability is not supported; Core::X86::Msr::PPIN_CTL and Core::X86::Msr::PPIN are treated as RAZ. 1=Indicates that Protected Processor Inventory Number (PPIN) capability can be enabled for privileged system inventory agent to read PPIN from Core::X86::Msr::PPIN. Protected Processor Inventory Number support.
22:21	Reserved.
20	<b>EferLmsleUnsupported.</b> Read-only. Reset: Fixed,1. 1=Core::X86::Msr::EFER[LMSLE] is not supported, and MBZ.
19	<b>IbrsProvidesSameModeProtection.</b> Read-only. Reset: 1. IBRS provides Same Mode Protection.
18	<b>IbrsPreferred.</b> Read-only. Reset: 1. 1=IBRS is preferred over software solution.
17	<b>StibpAlwaysOn.</b> Read-only. Reset: 1. Single Thread Indirect Branch Prediction Mode has Enhanced Performance and may be left Always On.
16	Reserved.
15	<b>STIBP.</b> Read-only. Reset: 1. Single Thread Indirect Branch Prediction.
14	<b>IBRS.</b> Read-only. Reset: 1. Indirect Branch Restricted Speculation.
13	<b>INT_WBINVD.</b> Read-only. Reset: 1. Interruptible WBINVD,WBNOINVD.
12	<b>IBPB.</b> Read-only. Reset: 1. Indirect Branch Prediction Barrier.
11:10	Reserved.
9	<b>WBNOINVD.</b> Read-only. Reset: 1. WBNOINVD writes all modified cache lines in the internal caches of the processor back to memory leaving the line valid (clean) in the internal caches.
8	<b>MCOMMIT: memory commit.</b> Read-only. Reset: 0. Memory commit instruction support.
7	Reserved.
6	<b>MBE.</b> Read-only. Reset: Fixed,1. Memory Bandwidth Enforcement.
5	Reserved.
4	<b>RDPRU: read processor register at user level.</b> Read-only. Reset: Fixed,1. RDPRU instruction allows reading MPERF and APERF at user level.
3	<b>INVLPGb.</b> Read-only. Reset: Fixed,0. INVLPGb instruction broadcasts a TLB invalidate to all threads in the system.
2	<b>RstrFpErrPtrs.</b> Read-only. Reset: Fixed,1. 1=FXSAVE, XSAVE, FXSAVEOPT, XSAVEC, XSAVES always save error pointers and FXRSTOR, XRSTOR, XRSTORS always restore error pointers is supported.
1	<b>InstRetCntMsr: instructions retired count support.</b> Read-only. Reset: Fixed,1. 1=Core::X86::Msr::IRPerfCount supported.
0	<b>CLZERO: Clear Zero Instruction.</b> Read-only. Reset: Fixed,1. CLZERO instruction zero's out the 64 byte cache line specified in RAX. Note: CLZERO instruction operations are cache-line aligned and RAX[5:0] is ignored.



**CPUID\_Fn80000008\_ECX [Size Identifiers] (Core::X86::Cpuid::SizeId)**

Read-only.

This provides information about the number of threads supported by the processor.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000008\_ECX

Bits	Description
31:18	Reserved.
17:16	<b>PerfTscSize: performance time-stamp counter size.</b> Read-only. Reset: Fixed,0h.
15:12	<b>ApicIdSize: APIC ID size.</b> Read-only. Reset: Xh. The number of bits in the initial Core::X86::Apic::ApicId[ApicId] value that indicate thread ID within a package.
11:0	<b>NC: number of threads - 1.</b> Read-only. Reset: XXXh. The number of threads in the package is NC+1 (e.g., if NC=0, then there is one thread).

**CPUID\_Fn80000008\_EDX [Feature Extended Size Edx] (Core::X86::Cpuid::FeatureExtSizeEdx)**

Read-only. Reset: Fixed,0001\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000008\_EDX

Bits	Description
31:24	Reserved.
23:16	<b>RdpruMax.</b> Read-only. Reset: Fixed,01h. RDPRU Instruction max input supported.
15:0	Reserved.

**CPUID\_Fn8000000A\_EAX [SVM Revision and Feature Identification] (Core::X86::Cpuid::SvmRevFeatIdEax)**

Read-only. Reset: Fixed,0000\_0001h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000000A\_EAX

Bits	Description
31:8	Reserved.
7:0	<b>SvmRev.</b> Read-only. Reset: Fixed,01h. SVM revision.

**CPUID\_Fn8000000A\_EBX [SVM Revision and Feature Identification] (Core::X86::Cpuid::SvmRevFeatIdEbx)**

Read-only, Volatile. Reset: 0000\_8000h.

This provides SVM revision and feature information.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000000A\_EBX

Bits	Description
31:0	<b>NASID: number of address space identifiers (ASID).</b> Read-only, Volatile. Reset: 0000_8000h.

**CPUID\_Fn8000000A\_EDX [SVM Revision and Feature Identification] (Core::X86::Cpuid::SvmRevFeatIdEdx)**

Read-only.

This provides SVM feature information.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000000A\_EDX

Bits	Description
31	<b>EnhancedShutdownIntercept.</b> Read-only. Reset: Fixed,1. Support for EXITINFO1 on shutdown intercept and nested shutdown intercepts will result in a non-interceptible shutdown.
30	<b>IdleHltIntercept.</b> Read-only. Reset: Fixed,1. Indicates support for intercepting HLT conditionally if there is no virtual interrupt pending
29	<b>GuestBusLockThreshold.</b> Read-only. Reset: Fixed,1. Allows hypervisor to intercept a guest after it has executed a programmable number of bus locks.
28	<b>VmcbAddrChkChg.</b> Read-only. Reset: Fixed,1. VMCB address check change
27	<b>ExtLvtOffsetFaultChg.</b> Read-only. Reset: Fixed,1. 1=Read/Write fault behavior for the extended LVT offsets (APIC addresses 0x500-0x530) changed to Read Allowed, Write #MVEXIT (trap).
26	<b>IbsVirt.</b> Read-only. Reset: Fixed,1. IBS Virtualization
25	<b>NmiVirt.</b> Read-only. Reset: Fixed,1. Guest NMI Virtualization
24	Reserved.
23	<b>HOST_MCE_OVERRIDE.</b> Read-only. Reset: Fixed,1. 1=If hCR4:MCE == 1 and gCR4:MCE == 0, machine check exceptions (#MC) in guest do not cause shutdown and are always intercepted.
22	Reserved.
21	<b>AllowNonWriteAbleGPT.</b> Read-only. Reset: Fixed,1. 1=Indicates support for Non-Writeable Guest Pages for NPT.
20	<b>GuestSpecCtrl.</b> Read-only. Reset: Fixed,1. 1=Indicates support for Guest Spec_ctl.
19	<b>SupervisorShadowStack.</b> Read-only. Reset: Fixed,1.
18	<b>X2AVIC: virtualized X2APIC.</b> Read-only. Reset: 1. 1=Virtualized X2APIC is supported.
17	<b>GMET.</b> Read-only. Reset: Fixed,1. Guest Mode Execute Trap.
16	<b>vGIF.</b> Read-only. Reset: Fixed,1. Virtualized GIF.
15	<b>V_VMSAVE_VMLOAD.</b> Read-only. Reset: Fixed,1. Virtualized VMLOAD and VMSAVE.
14	Reserved.
13	<b>AVIC: AMD virtual interrupt controller.</b> Read-only. Reset: 1. 1=Support indicated for SVM mode virtualized interrupt controller; Indicates support for Core::X86::Msr::AvicDoorbell.
12	<b>PauseFilterThreshold.</b> Read-only. Reset: Fixed,1.
11	Reserved.
10	<b>PauseFilter.</b> Read-only. Reset: Fixed,1. Pause intercept filter.
9	Reserved.
8	<b>PerfCtrVirt.</b> Read-only. Reset: Fixed,1. Indicates support for virtualization of Performance Monitor Counters.
7	<b>DecodeAssists.</b> Read-only. Reset: Fixed,1.
6	<b>FlushByAsid.</b> Read-only. Reset: Fixed,1.
5	<b>VmcbClean.</b> Read-only. Reset: Fixed,1. VMCB clean bits.
4	<b>TscRateMsr: MSR based TSC rate control.</b> Read-only. Reset: Fixed,1. 1=Indicates support for TSC ratio Core::X86::Msr::TscRateMsr. MSR based TSC rate control.
3	<b>NRIPS.</b> Read-only. Reset: Fixed,1. NRIP Save.
2	<b>SVML.</b> Read-only. Reset: Fixed,1. SVM lock.
1	<b>LbrVirt.</b> Read-only. Reset: Fixed,1. LBR virtualization.
0	<b>NP.</b> Read-only. Reset: Fixed,1. Nested Paging.

**CPUID\_Fn80000019\_EAX [L1 TLB 1G Identifiers] (Core::X86::Cpuid::L1Tlb1G)**

Read-only.

This function provides first level TLB characteristics for 1GB pages.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000019\_EAX

Bits	Description
31:28	<b>L1DTlb1GAssoc</b> : L1 data TLB associativity for 1 GB pages. Read-only. Reset: Fixed,Fh. See Core::X86::Cpuid::L2CacheId[L2Assoc].
27:16	<b>L1DTlb1GSize</b> . Read-only. Reset: Fixed,96. L1 data TLB number of entries for 1 GB pages.
15:12	<b>L1ITlb1GAssoc</b> . Read-only. Reset: Fixed,Fh. L1 instruction TLB associativity for 1 GB pages. See Core::X86::Cpuid::L2CacheId[L2Assoc].
11:0	<b>L1ITlb1GSize</b> . Read-only. Reset: Fixed,64. L1 instruction TLB number of entries for 1 GB pages.

**CPUID\_Fn80000019\_EBX [L2 TLB 1G Identifiers] (Core::X86::Cpuid::L2Tlb1G)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000019\_EBX

Bits	Description
31:28	<b>L2DTlb1GAssoc</b> . Read-only. Reset: Fixed,4. L2 data TLB associativity for 1 GB pages.
27:16	<b>L2DTlb1GSize</b> . Read-only. Reset: Fixed,020h. L2 data TLB number of entries for 1 GB pages.
15:12	<b>L2ITlb1GAssoc</b> . Read-only. Reset: Fixed,0h. L2 instruction TLB associativity for 1 GB pages.
11:0	<b>L2ITlb1GSize</b> . Read-only. Reset: Fixed,000h. L2 instruction TLB number of entries for 1 GB pages.

**CPUID\_Fn8000001A\_EAX [Performance Optimization Identifiers] (Core::X86::Cpuid::PerfOptId)**

Read-only. Reset: Fixed,0000\_000Ah.

This function returns performance related information.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001A\_EAX

Bits	Description
31:4	Reserved.
3	<b>FP512</b> . Read-only. Reset: Fixed,1. When set, the internal FP/SIMD execution datapath is 512-bits wide.
2	<b>FP256</b> . Read-only. Reset: Fixed,0. When set, the internal FP/SIMD execution datapath is 256-bits wide.
1	<b>MOVU</b> . Read-only. Reset: Fixed,1. MOVU SSE instructions are more efficient and should be preferred to SSE MOVL/MOVH. MOVUPS is more efficient than MOVLPS/MOVHPS. MOVUPD is more efficient than MOVLPS/MOVHPS.
0	<b>FP128</b> . Read-only. Reset: Fixed,0. When set, the internal FP/SIMD execution datapath is 128-bits wide.

**CPUID\_Fn8000001B\_EAX [Instruction Based Sampling Identifiers] (Core::X86::Cpuid::IbsIdEax)**

Read-only. Reset: Fixed, 0008\_1BFFh.

This function returns IBS feature information.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001B\_EAX

Bits	Description
31:20	Reserved.
19	<b>IbsUpdtdDtlbStats.</b> Read-only. Reset: Fixed, 1. Simplified DTLB page size and miss reporting in IBS_OP_DATA3.
18:13	Reserved.
12	<b>IbsLoadLatencyFiltering.</b> Read-only. Reset: Fixed, 1. Indicates support for filtering of IBS execution samples based on load latency.
11	<b>Zen4IbsExtensions.</b> Read-only. Reset: Fixed, 1. Fetch and Op IBS support IBS extensions added with Zen4: Core::X86::Msrr::IBS_FETCH_CTL[IbsFetchL3Miss], Core::X86::Msrr::IBS_FETCH_CTL[IbsFetchOcMiss], Core::X86::Msrr::IBS_FETCH_CTL[IbsL3MissOnly] and Core::X86::Msrr::IBS_OP_DATA2 DataSrc extension.
10	<b>IbsOpData4.</b> Read-only. Reset: Fixed, 0. IBS op data 4 MSR supported.
9	<b>IbsFetchCtlExtd: IBS fetch control extended MSR supported.</b> Read-only. Reset: Fixed, 1. Indicates support for Core::X86::Msrr::IC_IBS_EXTD_CTL.
8	<b>OpFuse: fused branch op indication supported.</b> Read-only. Reset: Fixed, 1. Indicates support for Core::X86::Msrr::IBS_OP_DATA[IbsOpFuse].
7	<b>RipInvalidChk: invalid RIP indication supported.</b> Read-only. Reset: Fixed, 1. Indicates support for Core::X86::Msrr::IBS_OP_DATA[IbsRipInvalid].
6	<b>OpCntExt: IbsOpCurCnt and IbsOpMaxCnt extend by 7 bits.</b> Read-only. Reset: Fixed, 1. Indicates support for Core::X86::Msrr::IBS_OP_CTL[IbsOpCurCnt[26:20], IbsOpMaxCnt[26:20]].
5	<b>BrnTrgt.</b> Read-only. Reset: Fixed, 1. Branch target address reporting supported.
4	<b>OpCnt.</b> Read-only. Reset: Fixed, 1. Op counting mode supported.
3	<b>RdWrOpCnt.</b> Read-only. Reset: Fixed, 1. Read write of op counter supported.
2	<b>OpSam.</b> Read-only. Reset: Fixed, 1. IBS execution sampling supported.
1	<b>FetchSam.</b> Read-only. Reset: Fixed, 1. IBS fetch sampling supported.
0	<b>IBSFFV.</b> Read-only. Reset: Fixed, 1. IBS feature flags valid.

**CPUID\_Fn8000001D\_EAX\_x00 [Cache Properties (DC)] (Core::X86::Cpuid::CachePropEax0)**

Core::X86::Cpuid::CachePropEax0 reports topology information for the DC.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_EAX\_x00

Bits	Description												
31:26	Reserved.												
25:14	<b>NumSharingCache: number of logical processors sharing cache.</b> Read-only. Reset: XXXh. The number of logical processors sharing this cache is NumSharingCache + 1.												
13:10	Reserved.												
9	<b>FullyAssociative: fully associative cache.</b> Read-only. Reset: Fixed,0. 1=Cache is fully associative.												
8	<b>SelfInitialization: cache is self-initializing.</b> Read-only. Reset: Fixed,1. 1=Cache is self initializing; cache does not need software initialization.												
7:5	<b>CacheLevel: cache level.</b> Read-only. Reset: Fixed,1h. Identifies the cache level. <b>ValidValues:</b>												
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Reserved.</td></tr> <tr> <td>1h</td><td>Level 1</td></tr> <tr> <td>2h</td><td>Level 2</td></tr> <tr> <td>3h</td><td>Level 3</td></tr> <tr> <td>7h-4h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	Reserved.	1h	Level 1	2h	Level 2	3h	Level 3	7h-4h	Reserved.
Value	Description												
0h	Reserved.												
1h	Level 1												
2h	Level 2												
3h	Level 3												
7h-4h	Reserved.												
4:0	<b>CacheType: cache type.</b> Read-only. Reset: Fixed,01h. Identifies the type of cache. <b>ValidValues:</b>												
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00h</td><td>Null; no more caches.</td></tr> <tr> <td>01h</td><td>Data cache.</td></tr> <tr> <td>02h</td><td>Instruction cache.</td></tr> <tr> <td>03h</td><td>Unified cache.</td></tr> <tr> <td>1Fh-04h</td><td>Reserved.</td></tr> </table>	Value	Description	00h	Null; no more caches.	01h	Data cache.	02h	Instruction cache.	03h	Unified cache.	1Fh-04h	Reserved.
Value	Description												
00h	Null; no more caches.												
01h	Data cache.												
02h	Instruction cache.												
03h	Unified cache.												
1Fh-04h	Reserved.												

**CPUID\_Fn8000001D\_EAX\_x01 [Cache Properties (IC)] (Core::X86::Cpuid::CachePropEax1)**

Read-only.

Core::X86::Cpuid::CachePropEax1 reports topology information for the IC. See Core::X86::Cpuid::CachePropEax0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_EAX\_x01

Bits	Description
31:26	Reserved.
25:14	<b>NumSharingCache: number of logical processors sharing cache.</b> Read-only. Reset: XXXh. See Core::X86::Cpuid::CachePropEax0[NumSharingCache].
13:10	Reserved.
9	<b>FullyAssociative: fully associative cache.</b> Read-only. Reset: Fixed,0. See Core::X86::Cpuid::CachePropEax0[FullyAssociative].
8	<b>SelfInitialization: cache is self-initializing.</b> Read-only. Reset: Fixed,1. See Core::X86::Cpuid::CachePropEax0[SelfInitialization].
7:5	<b>CacheLevel: cache level.</b> Read-only. Reset: Fixed,1h. Identifies the cache level. See Core::X86::Cpuid::CachePropEax0[CacheLevel].
4:0	<b>CacheType: cache type.</b> Read-only. Reset: Fixed,02h. See Core::X86::Cpuid::CachePropEax0[CacheType].

**CPUID\_Fn8000001D\_EAX\_x02 [Cache Properties (L2)] (Core::X86::Cpuid::CachePropEax2)**

Read-only.

Core::X86::Cpuid::CachePropEax2 reports topology information for the L2. See Core::X86::Cpuid::CachePropEax0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_EAX\_x02

Bits	Description
31:26	Reserved.
25:14	<b>NumSharingCache: number of logical processors sharing cache.</b> Read-only. Reset: XXXh. Core::X86::Cpuid::CachePropEax0[NumSharingCache].
13:10	Reserved.
9	<b>FullyAssociative: fully associative cache.</b> Read-only. Reset: Fixed,0. Core::X86::Cpuid::CachePropEax0[FullyAssociative].
8	<b>SelfInitialization: cache is self-initializing.</b> Read-only. Reset: Fixed,1. Core::X86::Cpuid::CachePropEax0[SelfInitialization].
7:5	<b>CacheLevel: cache level.</b> Read-only. Reset: Fixed,2h. Identifies the cache level. Core::X86::Cpuid::CachePropEax0[CacheLevel].
4:0	<b>CacheType: cache type.</b> Read-only. Reset: Fixed,03h. Core::X86::Cpuid::CachePropEax0[CacheType].

**CPUID\_Fn8000001D\_EAX\_x03 [Cache Properties (L3)] (Core::X86::Cpuid::CachePropEax3)**

Read-only.

Core::X86::Cpuid::CachePropEax3 reports topology information for the L3.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_EAX\_x03

Bits	Description
31:26	Reserved.
25:14	<b>NumSharingCache: number of logical processors sharing cache.</b> Read-only. Reset: XXXh. The number of logical processors sharing this cache is NumSharingCache + 1.
13:10	Reserved.
9	<b>FullyAssociative: fully associative cache.</b> Read-only. Reset: Fixed,0. Core::X86::Cpuid::CachePropEax0[FullyAssociative].
8	<b>SelfInitialization: cache is self-initializing.</b> Read-only. Reset: Fixed,1. Core::X86::Cpuid::CachePropEax0[SelfInitialization].
7:5	<b>CacheLevel: cache level.</b> Read-only. Reset: Fixed,3h. Identifies the cache level. Core::X86::Cpuid::CachePropEax0[CacheLevel].
4:0	<b>CacheType: cache type.</b> Read-only. Reset: Fixed,03h. Core::X86::Cpuid::CachePropEax0[CacheType].

**CPUID\_Fn8000001D\_EAX\_x04 [Cache Properties Null] (Core::X86::Cpuid::CachePropEax4)**

Read-only. Reset: Fixed,0000\_0000h.

Core::X86::Cpuid::CachePropEax4 reports done/null. See Core::X86::Cpuid::CachePropEax0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_EAX\_x04

Bits	Description
31:5	Reserved.
4:0	<b>CacheType: cache type.</b> Read-only. Reset: Fixed,00h. Core::X86::Cpuid::CachePropEax0[CacheType].

**CPUID\_Fn8000001D\_EBX\_x00 [Cache Properties (DC)] (Core::X86::Cpuid::CachePropEbx0)**

Read-only. Reset: Fixed,02C0\_003Fh.

Core::X86::Cpuid::CachePropEbx0 reports topology information for the DC. See Core::X86::Cpuid::CachePropEax0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_EBX\_x00

Bits	Description
31:22	<b>CacheNumWays: cache number of ways.</b> Read-only. Reset: Fixed,00Bh. Cache number of ways is CacheNumWays + 1.
21:12	<b>CachePhysPartitions: cache physical line partitions.</b> Read-only. Reset: Fixed,000h. Cache partitions is CachePhysPartitions + 1.
11:0	<b>CacheLineSize: cache line size in bytes.</b> Read-only. Reset: Fixed,03Fh. Cache line size in bytes is CacheLineSize + 1.

**CPUID\_Fn8000001D\_EBX\_x01 [Cache Properties (IC)] (Core::X86::Cpuid::CachePropEbx1)**

Read-only. Reset: Fixed,01C0\_003Fh.

Core::X86::Cpuid::CachePropEbx1 reports topology information for the IC. See Core::X86::Cpuid::CachePropEax0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_EBX\_x01

Bits	Description
31:22	<b>CacheNumWays: cache number of ways.</b> Read-only. Reset: Fixed,007h. Core::X86::Cpuid::CachePropEbx0[CacheNumWays].
21:12	<b>CachePhysPartitions: cache physical line partitions.</b> Read-only. Reset: Fixed,000h. Core::X86::Cpuid::CachePropEbx0[CachePhysPartitions].
11:0	<b>CacheLineSize: cache line size in bytes.</b> Read-only. Reset: Fixed,03Fh. Core::X86::Cpuid::CachePropEbx0[CacheLineSize].

**CPUID\_Fn8000001D\_EBX\_x02 [Cache Properties (L2)] (Core::X86::Cpuid::CachePropEbx2)**

Read-only. Reset: Fixed,03C0\_003Fh.

Core::X86::Cpuid::CachePropEbx2 reports topology information for the L2. See Core::X86::Cpuid::CachePropEax0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_EBX\_x02

Bits	Description
31:22	<b>CacheNumWays: cache number of ways.</b> Read-only. Reset: Fixed,00Fh. See Core::X86::Cpuid::CachePropEbx0[CacheNumWays].
21:12	<b>CachePhysPartitions: cache physical line partitions.</b> Read-only. Reset: Fixed,000h. See Core::X86::Cpuid::CachePropEbx0[CachePhysPartitions].
11:0	<b>CacheLineSize: cache line size in bytes.</b> Read-only. Reset: Fixed,03Fh. See Core::X86::Cpuid::CachePropEbx0[CacheLineSize].

**CPUID\_Fn8000001D\_EBX\_x03 [Cache Properties (L3)] (Core::X86::Cpuid::CachePropEbx3)**

Read-only. Reset: Fixed,03C0\_003Fh.

Core::X86::Cpuid::CachePropEbx3 reports topology information for the L3. See Core::X86::Cpuid::CachePropEax0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_EBX\_x03

Bits	Description
31:22	<b>CacheNumWays: cache number of ways.</b> Read-only. Reset: Fixed,00Fh. See Core::X86::Cpuid::CachePropEbx0[CacheNumWays].
21:12	<b>CachePhysPartitions: cache physical line partitions.</b> Read-only. Reset: Fixed,000h. See Core::X86::Cpuid::CachePropEbx0[CachePhysPartitions].
11:0	<b>CacheLineSize: cache line size in bytes.</b> Read-only. Reset: Fixed,03Fh. See Core::X86::Cpuid::CachePropEbx0[CacheLineSize].



**CPUID\_Fn8000001D\_EBX\_x04 [Cache Properties Null] (Core::X86::Cpuid::CachePropEbx4)**

Read-only. Reset: Fixed,0000\_0000h.

Core::X86::Cpuid::CachePropEax4 reports done/null. See Core::X86::Cpuid::CachePropEax0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_EBX\_x04

**Bits Description**

31:0 Reserved.

**CPUID\_Fn8000001D\_ECX\_x00 [Cache Properties (DC)] (Core::X86::Cpuid::CachePropEcxc0)**

Read-only. Reset: Fixed,0000\_003Fh.

Core::X86::Cpuid::CachePropEcxc0 reports topology information for the DC. See Core::X86::Cpuid::CachePropEax0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_ECX\_x00

**Bits Description**31:0 **CacheNumSets: cache number of sets.** Read-only. Reset: Fixed,0000\_003Fh. Cache number of sets is CacheNumSets + 1.**CPUID\_Fn8000001D\_ECX\_x01 [Cache Properties (IC)] (Core::X86::Cpuid::CachePropEcxc1)**

Read-only. Reset: Fixed,0000\_003Fh.

Core::X86::Cpuid::CachePropEcxc1 reports topology information for the IC. See Core::X86::Cpuid::CachePropEax0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_ECX\_x01

**Bits Description**31:0 **CacheNumSets: cache number of sets.** Read-only. Reset: Fixed,0000\_003Fh. See Core::X86::Cpuid::CachePropEcxc0[CacheNumSets].**CPUID\_Fn8000001D\_ECX\_x02 [Cache Properties (L2)] (Core::X86::Cpuid::CachePropEcxc2)**

Read-only. Reset: Fixed,0000\_03FFh.

Core::X86::Cpuid::CachePropEcxc2 reports topology information for the L2. See Core::X86::Cpuid::CachePropEax0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_ECX\_x02

**Bits Description**31:0 **CacheNumSets: cache number of sets.** Read-only. Reset: Fixed,0000\_03FFh. See Core::X86::Cpuid::CachePropEcxc0[CacheNumSets].



**CPUID\_Fn8000001D\_ECX\_x03 [Cache Properties (L3)] (Core::X86::Cpuid::CachePropEc3)**

Read-only.

Core::X86::Cpuid::CachePropEc3 reports topology information for the L3.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_ECX\_x03

Bits	Description																
31:0	<b>CacheNumSets: cache number of sets.</b> Read-only. Reset: 0000_XXXXh. See Core::X86::Cpuid::CachePropEc0[CacheNumSets]. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0000_1 FFEh- 0000_0 000h</td><td>Reserved.</td></tr> <tr> <td>0000_1 FFFh</td><td>8192 L3 Cache Sets.</td></tr> <tr> <td>0000_3 FFEh- 0000_2 000h</td><td>Reserved.</td></tr> <tr> <td>0000_3 FFFh</td><td>16384 L3 Cache Sets.</td></tr> <tr> <td>0000_7 FFEh- 0000_4 000h</td><td>Reserved.</td></tr> <tr> <td>0000_7 FFFh</td><td>32768 L3 Cache Sets.</td></tr> <tr> <td>FFFF_F FFFh- 0000_8 000h</td><td>Reserved.</td></tr> </table>	Value	Description	0000_1 FFEh- 0000_0 000h	Reserved.	0000_1 FFFh	8192 L3 Cache Sets.	0000_3 FFEh- 0000_2 000h	Reserved.	0000_3 FFFh	16384 L3 Cache Sets.	0000_7 FFEh- 0000_4 000h	Reserved.	0000_7 FFFh	32768 L3 Cache Sets.	FFFF_F FFFh- 0000_8 000h	Reserved.
Value	Description																
0000_1 FFEh- 0000_0 000h	Reserved.																
0000_1 FFFh	8192 L3 Cache Sets.																
0000_3 FFEh- 0000_2 000h	Reserved.																
0000_3 FFFh	16384 L3 Cache Sets.																
0000_7 FFEh- 0000_4 000h	Reserved.																
0000_7 FFFh	32768 L3 Cache Sets.																
FFFF_F FFFh- 0000_8 000h	Reserved.																

**CPUID\_Fn8000001D\_ECX\_x04 [Cache Properties Null] (Core::X86::Cpuid::CachePropEc4)**

Read-only. Reset: Fixed,0000\_0000h.

Core::X86::Cpuid::CachePropEax3 reports done/null. See Core::X86::Cpuid::CachePropEax0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_ECX\_x04

Bits	Description
31:0	<b>CacheNumSets.</b> Read-only. Reset: Fixed,0000_0000h. Cache number of sets.

**CPUID\_Fn8000001D\_EDX\_x00 [Cache Properties (DC)] (Core::X86::Cpuid::CachePropEdx0)**

Read-only. Reset: Fixed,0000\_0000h.

Core::X86::Cpuid::CachePropEdx0 reports topology information for the DC. See Core::X86::Cpuid::CachePropEax0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_EDX\_x00

Bits	Description
31:2	Reserved.
1	<b>CacheInclusive: cache inclusive.</b> Read-only. Reset: Fixed,0. 0=Cache is not inclusive of lower cache levels. 1=Cache is inclusive of lower cache levels.
0	<b>WBINVD: Write-Back Invalidate/Invalidate.</b> Read-only. Reset: Fixed,0. 0=WBINVD/INVD invalidates all lower level caches of non-originating cores sharing this cache. 1=WBINVD/INVD not ensured to invalidate all lower level caches of non-originating cores sharing this cache.

**CPUID\_Fn8000001D\_EDX\_x01 [Cache Properties (IC)] (Core::X86::Cpuid::CachePropEdx1)**

Read-only. Reset: Fixed,0000\_0000h.

Core::X86::Cpuid::CachePropEdx1 reports topology information for the IC. See Core::X86::Cpuid::CachePropEax0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_EDX\_x01

Bits	Description
31:2	Reserved.
1	<b>CacheInclusive: cache inclusive.</b> Read-only. Reset: Fixed,0. See Core::X86::Cpuid::CachePropEdx0[CacheInclusive].
0	<b>WBINVD: Write-Back Invalidate/Invalidate.</b> Read-only. Reset: Fixed,0. 0=WBINVD/INVD invalidates all lower level caches of non-originating cores sharing this cache. 1=WBINVD/INVD may not invalidate all lower level caches of non-originating cores sharing this cache. See Core::X86::Cpuid::CachePropEdx0[WBINVD].

**CPUID\_Fn8000001D\_EDX\_x02 [Cache Properties (L2)] (Core::X86::Cpuid::CachePropEdx2)**

Read-only. Reset: Fixed,0000\_0002h.

Core::X86::Cpuid::CachePropEdx2 reports topology information for the L2. See Core::X86::Cpuid::CachePropEax0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_EDX\_x02

Bits	Description
31:2	Reserved.
1	<b>CacheInclusive: cache inclusive.</b> Read-only. Reset: Fixed,1. See Core::X86::Cpuid::CachePropEdx0[CacheInclusive].
0	<b>WBINVD: Write-Back Invalidate/Invalidate.</b> Read-only. Reset: Fixed,0. 0=WBINVD/INVD invalidates all lower level caches of non-originating cores sharing this cache. 1=WBINVD/INVD may not invalidate all lower level caches of non-originating cores sharing this cache.

**CPUID\_Fn8000001D\_EDX\_x03 [Cache Properties (L3)] (Core::X86::Cpuid::CachePropEdx3)**

Read-only. Reset: Fixed,0000\_0001h.

Core::X86::Cpuid::CachePropEdx3 reports reports topology information for the L3. See

Core::X86::Cpuid::CachePropEax0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_EDX\_x03

Bits	Description
31:2	Reserved.
1	<b>CacheInclusive: cache inclusive.</b> Read-only. Reset: Fixed,0. See Core::X86::Cpuid::CachePropEdx0[CacheInclusive].
0	<b>WBINVD: Write-Back Invalidate/Invalidate.</b> Read-only. Reset: Fixed,1. 0=WBINVD/INVD invalidates all lower level caches of non-originating cores sharing this cache. 1=WBINVD/INVD may not invalidate all lower level caches of non-originating cores sharing this cache.

**CPUID\_Fn8000001D\_EDX\_x04 [Cache Properties Null] (Core::X86::Cpuid::CachePropEdx4)**

Read-only. Reset: Fixed,0000\_0000h.

Core::X86::Cpuid::CachePropEax3 reports done/null. See Core::X86::Cpuid::CachePropEax0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001D\_EDX\_x04

Bits	Description
31:0	Reserved.

**CPUID\_Fn8000001E\_EAX [Extended APIC ID] (Core::X86::Cpuid::ExtApicId)**

Read-only.

If Core::X86::Cpuid::FeatureExtIdEcX[TopologyExtensions] == 0 then CPUID Fn8000001E\_E[D,C,B,A]X are reserved.  
 If (Core::X86::Msr::APIC\_BAR[ApicEn] == 0) then Core::X86::Cpuid::ExtApicId[ExtendedApicId] is Reserved.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001E\_EAX

Bits	Description
31:0	<b>ExtendedApicId: extended APIC ID.</b> Read-only. Reserved. Reset: (Core::X86::Msr::APIC_BAR[ApicEn] && Core::X86::Msr::APIC_BAR[x2ApicEn]) ? Core::X86::Msr::APIC_ID[ApicId[31:0]] : Core::X86::Msr::APIC_BAR[ApicEn] ? {00_0000h , Core::X86::Apic::ApicId[ApicId]} : 0000_0000h.

**CPUID\_Fn8000001E\_EBX [Core Identifiers] (Core::X86::Cpuid::CoreId)**

Read-only.

See Core::X86::Cpuid::ExtApicId.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001E\_EBX

Bits	Description
31:16	Reserved.
15:8	<b>ThreadsPerCore: threads per core.</b> Read-only. Reset: XXh. The number of threads per core is ThreadsPerCore+1.
7:0	<b>CoreId: core ID.</b> Read-only. Reset: XXh. Identifies the unique per-socket logical core unit ID.

**CPUID\_Fn8000001E\_ECX [Node Identifiers] (Core::X86::Cpuid::NodeId)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001E\_ECX

Bits	Description						
31:11	Reserved.						
10:8	<b>NodesPerProcessor.</b> Read-only. Reset: XXXb. Nodes per processor. <b>ValidValues:</b>						
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>1 node per processor.</td></tr> <tr> <td>7h-1h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	1 node per processor.	7h-1h	Reserved.
Value	Description						
0h	1 node per processor.						
7h-1h	Reserved.						
7:0	<b>NodeId.</b> Read-only. Reset: Fixed,XXh. Node ID.						

**CPUID\_Fn8000001F\_EAX [AMD Secure Encryption EAX] (Core::X86::Cpuid::SecureEncryptionEax)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001F\_EAX

Bits	Description
31	Reserved.
30	<b>HvInUseWrAllowed.</b> Read-only. Reset: Fixed,0. Writes to hypervisor-owned pages are allowed when marked InUse.
29:25	Reserved.
24	<b>VmsaRegProt.</b> Read-only. Reset: Fixed,0. VMSA Register Protection is supported.
23	<b>SegmentedRmp.</b> Read-only. Reset: Fixed,0. Segmented RMP is supported.
22	<b>GuestInterceptControl.</b> Read-only. Reset: Fixed,0. Guest Intercept Control feature is supported for SEV_ES guests.
21	<b>RmpReadInstruction.</b> Read-only. Reset: Fixed,0. RMPREAD instruction is supported.
20	<b>SevFeaturesOptinPmcVirt.</b> Read-only. Reset: Fixed,0. PMC virtualization is supported for SEV-ES guests via opt-in control in SEV_FEATURES.
19	<b>IbsVirtualization.</b> Read-only. Reset: Fixed,0. IBS state virtualization is supported for SEV-ES guests via opt-in control in SEV_FEATURES.
18	<b>VirtualTomMsr.</b> Read-only. Reset: Fixed,0. Virtual TOM MSR is supported.
17	<b>VmgexitParameter.</b> Read-only. Reset: Fixed,0. VmgexitParameter is supported in SEV_FEATURES.
16	<b>VTE: Virtual Transparent Encryption for SEV.</b> Read-only. Reset: Fixed,0. The Virtual Transparent Encryption feature can be enabled to force all memory accesses within an SEV guest to be encrypted with the guest's key. When enabled the hardware pretends that the C-bits for all guest mode accesses are 1 regardless of the actual guest page tables.
15	<b>PreventHostIBS.</b> Read-only. Reset: Fixed,0. Prevent host IBS for a SEV-ES guest.
14	<b>DebugStateSwap.</b> Read-only. Reset: Fixed,0. 1=DR0-3 and DR0-3_MASK can be saved/restored on world switches.
13	<b>AlternateInjection.</b> Read-only. Reset: Fixed,0. 1=SEV-ES guests can use an encrypted vmcb field for event injection.
12	<b>RestrictInjection.</b> Read-only. Reset: Fixed,0. 1=SEV-ES guests can refuse all event-injections except #HV.
11	<b>Req64BitHypervisor.</b> Read-only. Reset: Fixed,0. Require 64-Bit Hypervisor.
10	<b>CoherencyEnforced.</b> Read-only. Reset: Fixed,0. Hardware enforces cache coherency.
9	<b>TscAuxVirtualization.</b> Read-only. Reset: Fixed,0. Hardware virtualizes TSC_AUX.
8	<b>SecureTsc.</b> Read-only. Reset: Fixed,0. Support for Secure TSC.
7	<b>VmplSSS.</b> Read-only. Reset: 0. The Supervisor Shadow Stack bit is supported in the VMPL permission set.
6	<b>RMPQUERY.</b> Read-only. Reset: 0. The RMPQUERY instruction is supported.
5	<b>VMPL.</b> Read-only. Reset: Fixed,0. VMPL VM Permission Levels supported.
4	<b>SNP.</b> Read-only. Reset: Fixed,0. RMP table can be enabled to protect memory even from hypervisor.
3	<b>SevEs.</b> Read-only. Reset: Fixed,0. Secure Encrypted ES.
2	<b>VmPgFlush: VM Page Flush MSR support.</b> Read-only. Reset: Fixed,0. VMPAGE_FLUSH is no longer supported.
1	<b>SEV.</b> Read-only. Reset: Fixed,0. Secure Encrypted Virtualization supported.
0	<b>SME.</b> Read-only. Reset: Fixed,1. Secure Memory Encryption supported.

**CPUID\_Fn8000001F\_EBX [AMD Secure Encryption EBX] (Core::X86::Cpuid::SecureEncryptionEbx)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001F\_EBX

Bits	Description																		
31:16	Reserved.																		
15:12	<b>VmplSupported.</b> Read-only. Reset: Fixed, 0h. Number of VMPLs supported.																		
11:6	<b>MemEncryptPhysAddWidth.</b> Read-only. Reset: 000XXXb. Reduction of physical address space in bits when memory encryption is enabled (0 indicates no reduction). <b>ValidValues:</b>																		
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00h</td><td>Physical Address width is not reduced.</td></tr> <tr> <td>01h</td><td>Physical Address width is reduced by one.</td></tr> <tr> <td>02h</td><td>Physical Address width is reduced by two.</td></tr> <tr> <td>03h</td><td>Physical Address width is reduced by three.</td></tr> <tr> <td>04h</td><td>Physical Address width is reduced by four.</td></tr> <tr> <td>05h</td><td>Physical Address width is reduced by five.</td></tr> <tr> <td>06h</td><td>Physical Address width is reduced by six.</td></tr> <tr> <td>3Fh-07h</td><td>Reserved.</td></tr> </table>	Value	Description	00h	Physical Address width is not reduced.	01h	Physical Address width is reduced by one.	02h	Physical Address width is reduced by two.	03h	Physical Address width is reduced by three.	04h	Physical Address width is reduced by four.	05h	Physical Address width is reduced by five.	06h	Physical Address width is reduced by six.	3Fh-07h	Reserved.
Value	Description																		
00h	Physical Address width is not reduced.																		
01h	Physical Address width is reduced by one.																		
02h	Physical Address width is reduced by two.																		
03h	Physical Address width is reduced by three.																		
04h	Physical Address width is reduced by four.																		
05h	Physical Address width is reduced by five.																		
06h	Physical Address width is reduced by six.																		
3Fh-07h	Reserved.																		
5:0	<b>CBit.</b> Read-only. Reset: 33h. Page table bit number used to enable memory encryption.																		

**CPUID\_Fn8000001F\_ECX [AMD Secure Encryption ECX] (Core::X86::Cpuid::SecureEncryptionEcX)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001F\_ECX

Bits	Description
31:0	<b>NumEncryptedGuests.</b> Read-only. Reset: XXXX_XXXXh. Indicates the maximum ASID value that may be used for an SEV-enabled guest.

**CPUID\_Fn8000001F\_EDX [Minimum ASID] (Core::X86::Cpuid::SecureEncryptionEdx)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn8000001F\_EDX

Bits	Description
31:0	<b>MinimumSEVASID: Minimum SEV enabled, SEV-ES disabled ASID.</b> Read-only. Reset: 0000_000Xh. Indicates the minimum ASID value that must be used for an SEV-enabled, SEV-ES-disabled guest.

**CPUID\_Fn80000020\_EAX\_x00 [Platform QoS Extended Feature Identifiers] (Core::X86::Cpuid::AmdQosExtEax0)**

Read-only. Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000020\_EAX\_x00

Bits	Description
31:0	Reserved.

**CPUID\_Fn80000020\_EBX\_x00 [Platform QoS Extended Feature Identifiers]  
(Core::X86::Cpuid::AmdQosExtEbx0)**

Read-only. Reset: 0000\_007Eh.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000020\_EBX\_x00

Bits	Description
31:7	Reserved.
6	<b>CDMA_TO_MAX_CBM.</b> Read-only. Reset: 1. CDMA L3 Cache Isolation Support.
5	<b>ABMC.</b> Read-only. Reset: 1. Assignable Bandwidth Monitoring Counters..
4	<b>L3RR.</b> Read-only. Reset: 1. AMD L3 Range Reservation.
3	<b>EVT_CFG.</b> Read-only. Reset: 1. AMD Bandwidth Monitoring Event Configuration.
2	<b>SMBE.</b> Read-only. Reset: 1. Slow Memory Bandwidth Enforcement. See Core::X86::Cpuid::PqeBandwidthEax2 - Core::X86::Cpuid::PqeBandwidthEdx2.
1	<b>MBE.</b> Read-only. Reset: 1. Memory Bandwidth Enforcement.
0	Reserved.

**CPUID\_Fn80000020\_ECX\_x00 [Platform QoS Extended Feature Identifiers]  
(Core::X86::Cpuid::AmdQosExtEcx0)**

Read-only. Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000020\_ECX\_x00

Bits	Description
31:0	Reserved.

**CPUID\_Fn80000020\_EDX\_x00 [Platform QoS Extended Feature Identifiers]  
(Core::X86::Cpuid::AmdQosExtEdx0)**

Read-only. Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000020\_EDX\_x00

Bits	Description
31:0	Reserved.

**CPUID\_Fn80000020\_EAX\_x01 [Platform QoS Enforcement for Memory Bandwidth]  
(Core::X86::Cpuid::PqeBandwidthEax1)**

Read-only. Reset: 0000\_000Ch.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000020\_EAX\_x01

Bits	Description
31:0	<b>BW_LEN: QOS Memory Bandwidth Enforcement Limit Size.</b> Read-only. Reset: 0000_000Ch. Size of the QOS Memory Bandwidth Enforcement Limit.

**CPUID\_Fn80000020\_EBX\_x01 [Platform QoS Enforcement for Memory Bandwidth]  
(Core::X86::Cpuid::PqeBandwidthEbx1)**

Read-only. Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000020\_EBX\_x01

Bits	Description
31:0	Reserved.

**CPUID\_Fn80000020\_ECX\_x01 [Platform QoS Enforcement for Memory Bandwidth]  
(Core::X86::Cpuid::PqeBandwidthEcx1)**

Read-only. Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000020\_ECX\_x01

Bits	Description
31:0	Reserved.

**CPUID\_Fn80000020\_EDX\_x01 [Platform QoS Enforcement for Memory Bandwidth]  
(Core::X86::Cpuid::PqeBandwidthEdx1)**

Read-only. Reset: 0000\_000Fh.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000020\_EDX\_x01

Bits	Description
31:0	<b>NumClassService</b> . Read-only. Reset: 0000_000Fh. Number of classes of service.

**CPUID\_Fn80000020\_EAX\_x02 [Enforcement for Slow Memory Bandwidth]  
(Core::X86::Cpuid::PqeBandwidthEax2)**

Read-only. Reset: 0000\_000Ch.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000020\_EAX\_x02

Bits	Description
31:0	<b>BW_LEN</b> . Read-only. Reset: 0000_000Ch. Slow Memory Bandwidth Enforcement Bit Range Length.

**CPUID\_Fn80000020\_EBX\_x02 [Enforcement for Slow Memory Bandwidth]  
(Core::X86::Cpuid::PqeBandwidthEbx2)**

Read-only. Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000020\_EBX\_x02

Bits	Description
31:0	Reserved.

**CPUID\_Fn80000020\_ECX\_x02 [Enforcement for Slow Memory Bandwidth]  
(Core::X86::Cpuid::PqeBandwidthEcX2)**

Read-only. Reset: 0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000020\_ECX\_x02

Bits	Description
31:0	Reserved.

**CPUID\_Fn80000020\_EDX\_x02 [Enforcement for Slow Memory Bandwidth]  
(Core::X86::Cpuid::PqeBandwidthEdx2)**

Read-only. Reset: 0000\_000Fh.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000020\_EDX\_x02

Bits	Description
31:0	<b>COS_MAX</b> . Read-only. Reset: 0000_000Fh. Maximum Class Of Service (COS) number for Memory Bandwidth Enforcement.

**CPUID\_Fn80000020\_EBX\_x03 [Platform QoS Monitoring Bandwidth Event Configuration]  
(Core::X86::Cpuid::PqeBandwidthEvtCfgEbx3)**

Read-only. Reset: 0000\_0002h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000020\_EBX\_x03

Bits	Description
31:8	Reserved.
7:0	<b>EVT_NUM</b> . Read-only. Reset: 02h. Reports the number of bandwidth events that can be configured.

**CPUID\_Fn80000020\_ECX\_x03 [Platform QoS Monitoring Bandwidth Event Configuration]  
(Core::X86::Cpuid::PqeBandwidthEvtCfgEc3)**

Read-only. Reset: 0000\_007Fh.

Identifies the bandwidth sources that can be tracked.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000020\_ECX\_x03

Bits	Description
31:7	Reserved.
6	<b>L3CacheBwVicMon.</b> Read-only. Reset: 1. Dirty victims to all types of memory.
5	<b>L3CacheRmtSlowBwFillMon.</b> Read-only. Reset: 1. Reads from remote memory the system identifies as slow memory.
4	<b>L3CacheLclSlowBwFillMon.</b> Read-only. Reset: 1. Reads from local memory the system identifies as slow memory.
3	<b>L3CacheRmtBwNtWrMon.</b> Read-only. Reset: 1. Non-temporal writes to remote memory.
2	<b>L3CacheLclBwNtWrMon.</b> Read-only. Reset: 1. Non-temporal writes to local memory.
1	<b>L3CacheRmtBwFillMon.</b> Read-only. Reset: 1. Reads from remote DRAM memory.
0	<b>L3CacheLclBwFillMon.</b> Read-only. Reset: 1. Reads from local DRAM memory.

**CPUID\_Fn80000020\_EAX\_x05 [Assignable Bandwidth Monitoring Counters]  
(Core::X86::Cpuid::PqeBandwidthEax5)**

Read-only. Reset: 0000\_0014h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000020\_EAX\_x05

Bits	Description																
31:9	Reserved.																
8	<b>OverflowBit.</b> Read-only. Reset: 0. If set, indicates whether bit 61 in QM_CTR MSR is an overflow bit.																
7:0	<b>CounterSize.</b> Read-only. Reset: 14h. Encode counter width in the QM_CTR MSR offset from 24b. Several example values are enumerated below. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00h</td><td>Indicates that family / model stepping should be used to determine counter size.</td></tr> <tr> <td>01h</td><td>25-bit Counters.</td></tr> <tr> <td>13h-02h</td><td>Reserved.</td></tr> <tr> <td>14h</td><td>44-bit Counters.</td></tr> <tr> <td>24h-15h</td><td>Reserved.</td></tr> <tr> <td>25h</td><td>61-bit Counters.</td></tr> <tr> <td>FFh-26h</td><td>Reserved.</td></tr> </table>	Value	Description	00h	Indicates that family / model stepping should be used to determine counter size.	01h	25-bit Counters.	13h-02h	Reserved.	14h	44-bit Counters.	24h-15h	Reserved.	25h	61-bit Counters.	FFh-26h	Reserved.
Value	Description																
00h	Indicates that family / model stepping should be used to determine counter size.																
01h	25-bit Counters.																
13h-02h	Reserved.																
14h	44-bit Counters.																
24h-15h	Reserved.																
25h	61-bit Counters.																
FFh-26h	Reserved.																

**CPUID\_Fn80000020\_EBX\_x05 [Assignable Bandwidth Monitoring Counters]  
(Core::X86::Cpuid::PqeBandwidthEvtCfgEbx5)**

Read-only. Reset: 0000\_001Fh.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000020\_EBX\_x05

Bits	Description
31:16	Reserved.
15:0	<b>MAX_ABMC.</b> Read-only. Reset: 001Fh. Maximum Supported Assignable Bandwidth Monitoring Counter ID.



**CPUID\_Fn80000020\_ECX\_x05 [Assignable Bandwidth Monitoring Counters]  
(Core::X86::Cpuid::PqeBandwidthEvtCfgEc5)**

Read-only. Reset: 0000\_0001h.

Identifies the bandwidth sources that can be tracked.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000020\_ECX\_x05

Bits	Description
31:1	Reserved.
0	<b>SELECT_CLOS.</b> Read-only. Reset: 1. The user can configure individual bandwidth monitoring counters to measure Bandwidth consumed by a CLOS instead of an RMID.

**CPUID\_Fn80000021\_EAX [Extended Feature 2 EAX] (Core::X86::Cpuid::FeatureExt2Eax)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000021\_EAX

Bits	Description
31	<b>SRSO_MSR_FIX</b> . Read-only. Reset: Fixed,0. Indicates that software can set BP_CFG[4] to mitigate any other cases of Speculative Return Stack Overflow.
30	<b>SRSO_USER_KERNEL_NO</b> . Read-only. Reset: Fixed,1. Indicates that the CPU is not vulnerable to Speculative Return Stack Overflow at the user-kernel boundary.
29	<b>SRSO_NO</b> . Read-only. Reset: Fixed,0. Indicates that the CPU is not vulnerable to Speculative Return Stack Overflow.
28	<b>IBPB_BRTYPE</b> . Read-only. Reset: Fixed,1. Indicates Core::X86::Msr::PRED_CMD[IBPB] flushes all branch type predictions from the branch predictor.
27	<b>SBPB</b> . Read-only. Reset: Fixed,1. Indicates support for the Selective Branch Predictor Barrier.
26:25	Reserved.
24	<b>ERAPS</b> . Read-only. Reset: 1. Indicates support for enhanced return address predictor security.
23	Reserved.
22	<b>WL_CLASS_SUPPORT</b> . Read-only. Reset: 1. Indicates support for workload based heuristic feedback to OS for scheduling decisions.
21	<b>FP512_DOWNGRADE</b> . Read-only. Reset: 1. Indicates support for downgrading FP512 datapath to FP256.
20	<b>PREFETCHI</b> . Read-only. Reset: 1. Indicates support for IC prefetch.
19	<b>FAST_REP_SCASB</b> . Read-only. Reset: 1. Indicates support for Fast short REP SCASB.
18	<b>EPSF</b> . Read-only. Reset: 1. Enhanced Predictive Store Forwarding supported.
17	<b>GpOnUserCpuid</b> . Read-only. Reset: 1. Indicates support for #GP when executing CPUID at CPL > 0.
16	<b>OPCODE_0F017_RECLAIM</b> . Read-only. Reset: 1. Reserves opcode space 0F 01/7 for AMD use, returns illegal instruction for opcodes in this space.
15	<b>AMD_ERMSB</b> . Read-only. Reset: 1. Processor supports AMD implementation of Enhanced REP MOVSB/STOSB.
14	<b>L2TlbSizeX32</b> . Read-only. Reset: 1. Indicates that L2TLB sizes are encoded as multiples of 32.
13	<b>PrefetchCtlMsr</b> . Read-only. Reset: 1. 1=Prefetch control MSR supported. See Core::X86::Msr::PrefetchControl.
12	<b>PMC2PreciseRetire</b> . Read-only. Reset: 1. Indicates support for Core::X86::Msr::PERF_LEGACY_CTL2[PreciseRetire].
11	<b>FSRC</b> . Read-only. Reset: Fixed,1. Fast Short Repe Cmpsb supported.
10	<b>FSRS</b> . Read-only. Reset: Fixed,1. Fast Short Rep Stosb supported.
9	<b>NoSmmCtlMSR</b> . Read-only. Reset: Fixed,1. Indicates that MSRC001_0116 (SMM_CTL) is not present.
8	<b>AutomaticIBRS</b> . Read-only. Reset: Fixed,1. Indicates that Core::X86::Msr::EFER[AutomaticIBRSEn] can be set to automatically toggle IBRS protection when CPL changes.
7	<b>UpperAddressIgnore</b> . Read-only. Reset: Fixed,1. Indicates that Core::X86::Msr::EFER[UAIE] bit controls canonical check for upper address bits for certain types of accesses.
6	<b>NullSelectorClearsBase</b> . Read-only. Reset: 1. 1=Null Selector Clears Base. When this bit is set, a null segment load clears the segment base.
5:4	Reserved.
3	<b>SmmPgCfgLock</b> . Read-only. Reset: Fixed,1. 1=SMM paging configuration lock supported.
2	<b>LFenceAlwaysSerializing</b> . Read-only. Reset: Fixed,1. LFENCE is always serializing.
1	<b>FsGsKernelGsBaseNonSerializing</b> . Read-only. Reset: Fixed,1. WRMSR to Core::X86::Msr::FS_BASE, Core::X86::Msr::GS_BASE, and Core::X86::Msr::KernelGSbase are non-serializing.
0	<b>NoNestedDataBp</b> . Read-only. Reset: Fixed,1. New data-breakpoints are ignored while switching to data-breakpoint handler.

**CPUID\_Fn80000021\_EBX [Extended Feature 2 EBX] (Core::X86::Cpuid::FeatureExt2Ebx)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000021\_EBX

Bits	Description
31:24	Reserved.
23:16	<b>RapSize.</b> Read-only. Reset: 08h. Return Address Predictor size. RapSize x 8 is the minimum number of CALL instructions software needs to execute to flush the RAP. Note that the CALL instructions cannot be followed by RET instructions.
15:0	<b>MicrocodePatchSize.</b> Read-only. Reset: XXXXh. Reports the size of the Microcode patch in 16-byte multiples. If 0, the size of the patch is at most 5568 (15C0h) bytes.

**CPUID\_Fn80000022\_EAX [Extended Performance Monitoring and Debug EAX]****(Core::X86::Cpuid::ExtPerfMonAndDbgEax)**

Read-only. Reset: Fixed,0000\_0007h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000022\_EAX

Bits	Description
31:3	Reserved.
2	<b>LbrAndPmcFreeze.</b> Read-only. Reset: Fixed,1. 1=PMC and LBR freeze is supported in Core::X86::Msr::DBG_CTL_MSR.
1	<b>LbrExtV2.</b> Read-only. Reset: Fixed,1. 1=LBR extension Version 2 supported.
0	<b>PerfMonV2.</b> Read-only. Reset: Fixed,1. 1=Performance Monitoring Version 2 supported.

**CPUID\_Fn80000022\_EBX [Extended Performance Monitoring and Debug EBX]****(Core::X86::Cpuid::ExtPerfMonAndDbgEbx)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000022\_EBX

Bits	Description
31:24	Reserved.
23:16	<b>NumPerfCtrUmc.</b> Read-only. Reset: Fixed,XXXXXXXXXb. Number of available UMC PMCs.
15:10	<b>NumPerfCtrNB.</b> Read-only. Reset: Fixed,08h. Number of available Data Fabric (Northbridge) Performance Monitor Counters.
9:4	<b>LbrV2StackSz.</b> Read-only. Reset: Fixed,10h. Number of available LBR stack entries.
3:0	<b>NumPerfCtrCore.</b> Read-only. Reset: Fixed,6h. Number of Core Performance Counters.

**CPUID\_Fn80000022\_ECX [Extended Performance Monitoring and Debug ECX]****(Core::X86::Cpuid::ExtPerfMonAndDbgEcX)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000022\_ECX

Bits	Description
31:0	<b>ActiveUmcMask.</b> Read-only. Reset: Fixed,XXXX_XXXXh. Bitmask representing active UMCs. Calculate the number of PMCs per UMC as Core::X86::Cpuid::ExtPerfMonAndDbgEbx[NumPerfCtrUmc] / POPCNT(CPUID_8000_0022_ECX[31:0]).

**CPUID\_Fn80000023\_EAX [AMD Secure Multi-Key Encryption EAX]****(Core::X86::Cpuid::SecureMultiKeyEncryptionEax)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000023\_EAX

Bits	Description
31:1	Reserved.
0	<b>MemHmkSupport.</b> Read-only. Reset: Fixed,0. 1=MEM-HMK mode is supported.

**CPUID\_Fn80000023\_EBX [AMD Secure Multi-Key Encryption EBX]  
(Core::X86::Cpuid::SecureMultiKeyEncryptionEbx)**

Read-only. Reset: Fixed,0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000023\_EBX

Bits	Description
31:16	Reserved.
15:0	<b>MaxMemHmkEncrKeyID.</b> Read-only. Reset: Fixed,0000h. Total number of available encryption keys in MEM-HMK mode.

**CPUID\_Fn80000023\_ECX [AMD Secure Multi-Key Encryption ECX]  
(Core::X86::Cpuid::SecureMultiKeyEncryptionEcx)**

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000023\_ECX

Bits	Description
31:0	Reserved.

**CPUID\_Fn80000023\_EDX [AMD Secure Multi-Key Encryption EDX]  
(Core::X86::Cpuid::SecureMultiKeyEncryptionEdx)**

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000023\_EDX

Bits	Description
31:0	Reserved.

**CPUID\_Fn80000026\_EAX\_x0[0...3] [Extended CPU Topology] (Core::X86::Cpuid::ExCpuTopologyEax)**

Read-only.

CPUID Fn8000\_0026\_E[D,C,B,A]X\_x[3:0] specifies the hierarchy of logical cores from the SMT level through the processor socket level. Software reads CPUID Fn8000\_0026\_E[C,B,A]X for ascending values of ECX until (CPUID Fn8000\_0026\_EBX[LogProcAtThisLevel] == 0).

Note: While CPUID Fn8000\_0026 is a preferred superset to CPUID\_Fn0000000B, CPUID\_Fn0000000B information is valid for software for the supported levels on AMD.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n0; CPUID\_Fn80000026\_EAX\_x00

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n1; CPUID\_Fn80000026\_EAX\_x01

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n2; CPUID\_Fn80000026\_EAX\_x02

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n3; CPUID\_Fn80000026\_EAX\_x03

Bits	Description
31	<b>AsymmetricCores.</b> Read-only. Reset: Fixed,X. When set, not all cores in the system report the same value in Core::X86::Cpuid::ExtCpuTopologyEbx[LogProcThisLevel] for Core::X86::Cpuid::ExtCpuTopologyEcx[LevelType]=0x1. When cleared, all cores in the system report the same value in Core::X86::Cpuid::ExtCpuTopologyEbx[LogProcThisLevel] for Core::X86::Cpuid::ExtCpuTopologyEcx[LevelType]=0x1.
30	<b>HeterogeneousCoreTopology.</b> Read-only. Reset: Fixed,X. When set, not all instances at the current hierarchy level have the same Core Type topology. The core type is reported in Core::X86::Cpuid::ExtCpuTopologyEbx[CoreType].
29	<b>EfficiencyRankingAvailable.</b> Read-only. Reset: Fixed,X. When set, Core::X86::Cpuid::ExtCpuTopologyEbx[ProcessorPowerEfficiencyRanking] is valid and varies between individual cores.
28:5	Reserved.
4:0	<b>CoreMaskWidth.</b> Read-only. Reset: Fixed,XXh. Number of bits to shift Core::X86::Cpuid::ExtCpuTopologyEdx[ExtendedLocalApicId] right to get unique topology ID of the next instance of the current level type. All logical processors with the same next level ID share current level.

**CPUID\_Fn80000026\_EBX\_x0[0...3] [Extended CPU Topology] (Core::X86::Cpuid::ExtCpuTopologyEbx)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n0; CPUID\_Fn80000026\_EBX\_x00

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n1; CPUID\_Fn80000026\_EBX\_x01

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n2; CPUID\_Fn80000026\_EBX\_x02

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n3; CPUID\_Fn80000026\_EBX\_x03

Bits	Description								
31:28	<b>CoreType.</b> Read-only. Reset: Fixed,Xh. Defines per-core architectural feature differentiation (microarchitectural resources, etc.) that may lead to a different performance, core clock boost, and power characteristic. Only valid while LevelType=Core. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Performance Core.</td></tr> <tr> <td>1h</td><td>Efficiency Core.</td></tr> <tr> <td>Fh-2h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	Performance Core.	1h	Efficiency Core.	Fh-2h	Reserved.
Value	Description								
0h	Performance Core.								
1h	Efficiency Core.								
Fh-2h	Reserved.								
27:24	<b>NativeModelId.</b> Read-only. Reset: Fixed,0h. Context sensitive to CPUID_Fn00000001_EAX and CPUID_Fn80000001_EAX [Family, Model, Stepping Identifiers] (Core::X86::Cpuid::FamModStep). Only valid while Level Type=Core. This value is used by software to further differentiate implementation specific software visible properties between the cores. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Zen5 Core.</td></tr> <tr> <td>Fh-1h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	Zen5 Core.	Fh-1h	Reserved.		
Value	Description								
0h	Zen5 Core.								
Fh-1h	Reserved.								
23:16	<b>ProcessorPowerEfficiencyRanking.</b> Read-only. Reset: Fixed,XXh. Identifies a static efficiency ranking between each of the cores of a specific CoreType, where a core with a lower value has intrinsically better power, but potentially lower performance potential vs cores with a higher value.								
15:0	<b>LogProcThisLevel.</b> Read-only. Reset: Fixed,XXXh. Number of logical processors at this level type.								

**CPUID\_Fn80000026\_ECX\_x0[0...3] [Extended CPU Topology] (Core::X86::Cpuid::ExtCpuTopologyEcX)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n0; CPUID\_Fn80000026\_ECX\_x00

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n1; CPUID\_Fn80000026\_ECX\_x01

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n2; CPUID\_Fn80000026\_ECX\_x02

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n3; CPUID\_Fn80000026\_ECX\_x03

Bits	Description														
31:16	Reserved.														
15:8	<b>LevelType.</b> Read-only. Reset: Fixed,XXh. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00h</td><td>Reserved.</td></tr> <tr> <td>01h</td><td>Core</td></tr> <tr> <td>02h</td><td>Complex</td></tr> <tr> <td>03h</td><td>Reserved.</td></tr> <tr> <td>04h</td><td>Socket</td></tr> <tr> <td>FFh-05h</td><td>Reserved.</td></tr> </table>	Value	Description	00h	Reserved.	01h	Core	02h	Complex	03h	Reserved.	04h	Socket	FFh-05h	Reserved.
Value	Description														
00h	Reserved.														
01h	Core														
02h	Complex														
03h	Reserved.														
04h	Socket														
FFh-05h	Reserved.														
7:0	<b>EcXVal.</b> Read-only. Reset: Fixed,XXh. ECX input value.														

**CPUID\_Fn80000026\_EDX [Extended CPU Topology] (Core::X86::Cpuid::ExtCpuTopologyEdx)**

Read-only.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000026\_EDX

**Bits Description**31:0 **ExtendedLocalApicId.** Read-only. Reset: Fixed,XXXX\_XXXXh. Extended APIC ID.**CPUID\_Fn80000027\_EAX [Hetero Workload Classification] (Core::X86::Cpuid::HeteroWorkloadClass)**

Read-only. Reset: 0000\_0003h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; CPUID\_Fn80000027\_EAX

**Bits Description**

31:4 Reserved.

3:0 **NUM\_WORK\_CLASS.** Read-only. Reset: 3h. Number of Workload Class ID's supported.**2.1.13 MSR Registers****2.1.13.1 MSRs - MSR0000\_xxxx****MSR0000\_0010 [Time Stamp Counter] (Core::X86::Msr::TSC)**

Read-write, Volatile. Reset: 0000\_0000\_0000\_0000h.

The TSC uses a common reference for all sockets, cores and threads.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0010

**Bits Description**63:0 **TSC: time stamp counter.** Read-write, Volatile. Reset: 0000\_0000\_0000\_0000h. The TSC increments at the P0 frequency. The TSC counts at the same rate in all P-states, all C states, S0, or S1. A Read of this MSR in guest mode is affected by Core::X86::Msr::TscRateMsr. The value (TSC/TSCRatio) is the TSC P0 frequency based value (as if TSCRatio == 1.0) when (TSCRatio != 1.0).**MSR0000\_001B [APIC Base Address] (Core::X86::Msr::APIC\_BAR)**

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_001B

**Bits Description**

63:48 Reserved.

47:12 **ApicBar[47:12]: APIC base address register.** Read-write. Reset: 0\_000F\_EE00h. Specifies the base address, physical address [47:12], for the APICXX register set in xAPIC mode. See 2.1.11.2.1.2 [APIC Register Space].11 **ApicEn: APIC enable.** Read-write. Reset: 0. 0=Disable Local APIC. 1=Local APIC is enabled in xAPIC mode. See 2.1.11.2.1.2 [APIC Register Space].10 **x2ApicEn: Extended APIC enable.** Read-write. Reset: 0. 0=Disable Extended Local APIC. 1=Extended Local APIC is enabled in x2APIC mode. Clearing this bit after it has been set requires ApicEn to be cleared as well.

9 Reserved.

8 **BSC: boot strap core.** Read-write, Volatile. Reset: X. 0=The core is not the boot core of the BSP. 1=The core is the boot core of the BSP.

7:0 Reserved.

**MSR0000\_002A [Cluster ID] (Core::X86::Msr::EBL\_CR\_POWERON)**

Writes to this register result in a GP fault with error code 0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_002A

**Bits Description**

63:18 Reserved.

17:16 **ClusterID.** Read, Error-on-write. Reset: 0h. The field does not affect hardware.

15:0 Reserved.

**MSR0000\_003B [Time Stamp Counter Adjustment] (Core::X86::Msr::TSC\_ADJUST)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_003B

Bits	Description
63:0	<b>TimeStampCounterAdjustment.</b> Read-write. Reset: 0000_0000_0000_0000h. Provides an offset to the TSC when the TSC is read. This value changes with a write, and does not change as time elapses.

**MSR0000\_0048 [Speculative Control] (Core::X86::Msr::SPEC\_CTRL)**

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0048

Bits	Description
63:8	Reserved.
7	<b>PSFD: Predictive Store Forwarding Disable.</b> Read-write. Reset: 0. 1=Disable predictive store forwarding.
6:3	Reserved.
2	<b>SSBD.</b> Read-write. Reset: 0. Speculative Store Bypass Disable.
1	<b>STIBP.</b> Read-write. Reset: 0. Single thread indirect branch predictor.
0	<b>IBRS.</b> Read-write. Reset: 0. Indirect branch restriction speculation.

**MSR0000\_0049 [Prediction Command] (Core::X86::Msr::PRED\_CMD)**

Write-only,Error-on-read. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]; MSR0000\_0049

Bits	Description
63:8	Reserved.
7	<b>SBPB: selective branch predictor barrier.</b> Write-only,Error-on-read. Reset: 0. When SBPB is supported (Core::X86::Cpuid::FeatureExt2Eax[SBPB]==1), setting this bit initiates a selective branch predictor barrier
6:1	Reserved.
0	<b>IBPB: indirect branch prediction barrier.</b> Write-only,Error-on-read. Reset: 0. Supported if Core::X86::Cpuid::FeatureExtIdEbx[IBPB] == 1.

**MSR0000\_008B [Patch Level] (Core::X86::Msr::PATCH\_LEVEL)**

Read,Error-on-write,Volatile. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]; MSR0000\_008B

Bits	Description
63:32	Reserved.
31:0	<b>PatchLevel.</b> Read,Error-on-write,Volatile. Reset: 0000_0000h. This returns an identification number for the microcode patch that has been loaded. If no patch has been loaded, this returns 0.

**MSR0000\_00E7 [Max Performance Frequency Clock Count] (Core::X86::Msr::MPERF)**

Read-write,Volatile. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_00E7

Bits	Description
63:0	<b>MPERF: maximum core clocks counter.</b> Read-write,Volatile. Reset: 0000_0000_0000_0000h. Incremented by hardware at the P0 frequency while the core is in C0. This register does not increment when the core is in the stop-grant state. In combination with Core::X86::Msr::APERF, this is used to determine the effective frequency of the core. A Read of this MSR in guest mode is affected by Core::X86::Msr::TscRateMsr. This field uses software P-state numbering. See Core::X86::Msr::HWCR[EffFreqCntMwait], 2.1.4 [Effective Frequency]



**MSR0000\_00E8 [Actual Performance Frequency Clock Count] (Core::X86::Msr::APERF)**

Read-write, Volatile. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_00E8

Bits	Description
63:0	<b>APERF: actual core clocks counter.</b> Read-write, Volatile. Reset: 0000_0000_0000_0000h. This register increments in proportion to the actual number of core clocks cycles while the core is in C0. The register does not increment when the core is in the stop-grant state. See Core::X86::Msr::MPERF.

**MSR0000\_00FE [MTRR Capabilities] (Core::X86::Msr::MTRRcap)**

Read, Error-on-write. Reset: 0000\_0000\_0000\_0508h.

\_lthree[1:0]\_core[7:0]; MSR0000\_00FE

Bits	Description
63:11	Reserved.
10	<b>MtrrCapWc: write-combining memory type.</b> Read, Error-on-write. Reset: 1. 1=The write combining memory type is supported.
9	Reserved.
8	<b>MtrrCapFix: fixed range register.</b> Read, Error-on-write. Reset: 1. 1=Fixed MTRRs are supported.
7:0	<b>MtrrCapVCnt: variable range registers count.</b> Read, Error-on-write. Reset: 08h. Specifies the number of variable MTRRs supported.

**MSR0000\_010B [Flush Command] (Core::X86::Msr::FLUSH\_CMD)**

Writes to this register do not execute until all prior instructions finished execution and have completed locally. Later instructions do not execute until the Write completes.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_010B

Bits	Description
63:1	Reserved.
0	<b>L1D_FLUSH.</b> Write-only, Volatile. Reset: 0. When written to 1, performs a write-back and invalidate of the L1 data cache.

**MSR0000\_0174 [SYSENTER CS] (Core::X86::Msr::SYSENTER\_CS)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0174

Bits	Description
63:16	Reserved.
15:0	<b>SysEnterCS: SYSENTER target CS.</b> Read-write. Reset: 0000h. Holds the called procedure code segment.

**MSR0000\_0175 [SYSENTER ESP] (Core::X86::Msr::SYSENTER\_ESP)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0175

Bits	Description
63:32	Reserved.
31:0	<b>SysEnterESP: SYSENTER target SP.</b> Read-write. Reset: 0000_0000h. Holds the called procedure stack pointer.

**MSR0000\_0176 [SYSENTER EIP] (Core::X86::Msr::SYSENTER\_EIP)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0176

Bits	Description
63:32	Reserved.
31:0	<b>SysEnterEIP: SYSENTER target IP.</b> Read-write. Reset: 0000_0000h. Holds the called procedure instruction pointer.



**MSR0000\_0179 [Global Machine Check Capabilities] (Core::X86::Msr::MCG\_CAP)**

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0179

Bits	Description
63:9	Reserved.
8	<b>McgCtlP: MCG_CTL register present.</b> Read-only, Error-on-write. Reset: Fixed, 1. 1=The machine check control registers (MCI_CTL) are present. See 3.1 [Machine Check Architecture].
7:0	<b>Count.</b> Read-only, Error-on-write, Volatile. Reset: XXh. Indicates the number of error reporting banks visible to the core. This value may differ from core to core.

**MSR0000\_017A [Global Machine Check Status] (Core::X86::Msr::MCG\_STAT)**

Read-write, Volatile. Reset: 0000\_0000\_0000\_0000h.

See 3.1 [Machine Check Architecture].

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_017A

Bits	Description
63:3	Reserved.
2	<b>MCIP.</b> Read-write, Volatile. Reset: 0. 1=A machine check is in progress. Machine check in progress.
1	<b>EIPV: error instruction pointer valid.</b> Read-write, Volatile. Reset: 0. 1=The instruction pointer that was pushed onto the stack by the machine check mechanism references the instruction that caused the machine check error.
0	<b>RIPV: restart instruction pointer valid.</b> Read-write, Volatile. Reset: 0. 0=The interrupt was not precise and/or the process (task) context may be corrupt; continued operation of this process may not be possible without intervention, however system processing or other processes may be able to continue with appropriate software clean up. 1=Program execution can be reliably restarted at the EIP address on the stack.

**MSR0000\_017B [Global Machine Check Exception Reporting Control] (Core::X86::Msr::MCG\_CTL)**

Reset: 0000\_0000\_0000\_0000h.

This register controls enablement of the individual error reporting banks; see 3.1 [Machine Check Architecture] and 3.1.2.1 [Global Registers]. When a machine check register bank is not enabled in MCG\_CTL, errors for that bank are not logged or reported, and actions enabled through the MCA are not taken; each MCI\_CTL register identifies which errors are still corrected when MCG\_CTL[i] is disabled.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_017B

Bits	Description																
63:7	<b>MCnEn.</b> Configurable. Reset: 000_0000_0000_0000h. <b>Description:</b> 1=The MC machine check register bank is enabled. Width of this field is SOC implementation and configuration specific. See 3.1.2.1 [Global Registers].																
6:0	<b>MCnEnCore.</b> Read-write. Reset: 00h. 1=The MC machine check register bank is enabled. <b>ValidValues:</b> <table border="1"> <tr> <th>Bit</th><th>Description</th></tr> <tr> <td>[0]</td><td>Enable MCA for LSDC</td></tr> <tr> <td>[1]</td><td>Enable MCA for ICBP</td></tr> <tr> <td>[2]</td><td>Enable MCA for L2</td></tr> <tr> <td>[3]</td><td>Enable MCA for DE</td></tr> <tr> <td>[4]</td><td>Reserved.</td></tr> <tr> <td>[5]</td><td>Enable MCA for SCEX</td></tr> <tr> <td>[6]</td><td>Enable MCA for FP</td></tr> </table>	Bit	Description	[0]	Enable MCA for LSDC	[1]	Enable MCA for ICBP	[2]	Enable MCA for L2	[3]	Enable MCA for DE	[4]	Reserved.	[5]	Enable MCA for SCEX	[6]	Enable MCA for FP
Bit	Description																
[0]	Enable MCA for LSDC																
[1]	Enable MCA for ICBP																
[2]	Enable MCA for L2																
[3]	Enable MCA for DE																
[4]	Reserved.																
[5]	Enable MCA for SCEX																
[6]	Enable MCA for FP																

**MSR0000\_01D9 [Debug Control] (Core::X86::Msr::DBG\_CTL\_MSR)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_01D9

Bits	Description
63:13	Reserved.
12	<b>FPMCI</b> . Read-write. Reset: 0. 1=Freeze PMCs on PMC interrupt.
11	<b>FLBRI</b> . Read-write. Reset: 0. 1=Freeze LBR Stack on PMC interrupt.
10:6	Reserved.
5:3	<b>PB: performance monitor pin control</b> . Read-write. Reset: 0h. Performance Monitor Pins are not supported on this processor.
2	<b>BusLockTrapEn</b> . Read-write. Reset: 0.
1	<b>BTF</b> . Read-write. Reset: 0. 1=Enable branch single step.
0	<b>LBR</b> . Read-write. Reset: 0. 1=Enable last branch record.

**MSR0000\_01DB [Last Branch From IP] (Core::X86::Msr::BR\_FROM)**

Read,Error-on-write,Volatile. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_01DB

Bits	Description
63:58	Reserved.
57:0	<b>LastBranchFromIP</b> . Read,Error-on-write,Volatile. Reset: 000_0000_0000_0000h. Loaded with the segment offset of the branch instruction.

**MSR0000\_01DC [Last Branch To IP] (Core::X86::Msr::BR\_TO)**

Read,Error-on-write,Volatile. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_01DC

Bits	Description
63:58	Reserved.
57:0	<b>LastBranchToIP</b> . Read,Error-on-write,Volatile. Reset: 000_0000_0000_0000h. Holds the target RIP of the last branch that occurred before an exception or interrupt.

**MSR0000\_01DD [Last Exception From IP] (Core::X86::Msr::LastExcpFromIp)**

Read,Error-on-write,Volatile. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_01DD

Bits	Description
63:58	Reserved.
57:0	<b>LastIntFromIP</b> . Read,Error-on-write,Volatile. Reset: 000_0000_0000_0000h. Holds the source RIP of the last branch that occurred before the exception or interrupt.

**MSR0000\_01DE [Last Exception To IP] (Core::X86::Msr::LastExcpToIp)**

Read,Error-on-write,Volatile. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_01DE

Bits	Description
63:58	Reserved.
57:0	<b>LastIntToIP</b> . Read,Error-on-write,Volatile. Reset: 000_0000_0000_0000h. Holds the target RIP of the last branch that occurred before the exception or interrupt.

**MSR0000\_020[0...E] [Variable-Size MTRRs Base] (Core::X86::Msr::MtrrVarBase)**

Each MTRR (Core::X86::Msr::MtrrVarBase, Core::X86::Msr::MtrrFix\_64K through Core::X86::Msr::MtrrFix\_4K\_7, or Core::X86::Msr::MTRRdefType) specifies a physical address range and a corresponding memory type (MemType) associated with that range. Setting the memory type to an unsupported value will result in a #GP.

The variable-size MTRRs come in pairs of base and mask registers (MSR0000\_0200 and MSR0000\_0201 are the first pair, etc.). Variables MTRRs are enabled through Core::X86::Msr::MTRRdefType[MtrrDefTypeEn]. A core access--with address CPUAddr--is determined to be within the address range of a variable-size MTRR if the following equation is true:

$\text{CPUAddr}[47:12] \& \text{PhyMask}[47:12] == \text{PhyBase}[47:12] \& \text{PhyMask}[47:12]$ .

For example, if the variable MTRR spans 256 KB and starts at the 1 MB address the PhyBase would be set to 0\_0010\_0000h and the PhyMask to F\_FFFC\_0000h (with zeros filling in for bits[11:0]). This results in a range from 0\_0010\_0000h to 0\_0013\_FFFFh.

\_lthree[1:0]\_core[7:0]\_n0; MSR0000\_0200

\_lthree[1:0]\_core[7:0]\_n1; MSR0000\_0202

\_lthree[1:0]\_core[7:0]\_n2; MSR0000\_0204

\_lthree[1:0]\_core[7:0]\_n3; MSR0000\_0206

\_lthree[1:0]\_core[7:0]\_n4; MSR0000\_0208

\_lthree[1:0]\_core[7:0]\_n5; MSR0000\_020A

\_lthree[1:0]\_core[7:0]\_n6; MSR0000\_020C

\_lthree[1:0]\_core[7:0]\_n7; MSR0000\_020E

Bits	Description
63:48	Reserved.
47:12	<b>PhyBase: base address.</b> Read-write. Reset: X_XXXX_XXXXh. Physical base address.
11:3	Reserved.
2:0	<b>MemType: memory type.</b> Read-write. Reset: XXXb. Address range from 00000h to 0FFFFh.
<b>ValidValues:</b>	
Value	Description
0h	UC or uncacheable.
1h	WC or write combining.
3h-2h	Reserved.
4h	WT or write through.
5h	WP or write protect.
6h	WB or write back.
7h	Reserved.

**MSR0000\_020[1...F] [Variable-Size MTRRs Mask] (Core::X86::Msr::MtrrVarMask)**

\_lthree[1:0]\_core[7:0]\_n0; MSR0000\_0201

\_lthree[1:0]\_core[7:0]\_n1; MSR0000\_0203

\_lthree[1:0]\_core[7:0]\_n2; MSR0000\_0205

\_lthree[1:0]\_core[7:0]\_n3; MSR0000\_0207

\_lthree[1:0]\_core[7:0]\_n4; MSR0000\_0209

\_lthree[1:0]\_core[7:0]\_n5; MSR0000\_020B

\_lthree[1:0]\_core[7:0]\_n6; MSR0000\_020D

\_lthree[1:0]\_core[7:0]\_n7; MSR0000\_020F

Bits	Description
63:48	Reserved.
47:12	<b>PhyMask: address mask.</b> Read-write. Reset: X_XXXX_XXXXh. Physical address mask.
11	<b>Valid: valid.</b> Read-write. Reset: X. 1=The variable-size MTRR pair is enabled.
10:0	Reserved.

**MSR0000\_0250 [Fixed-Size MTRRs] (Core::X86::Msr::MtrrFix\_64K)**

See Core::X86::Msr::MtrrVarBase for general MTRR information. Fixed MTRRs are enabled through Core::X86::Msr::MTRRdefType[MtrrDefTypeFixEn,MtrrDefTypeEn]. For addresses below 1MB, the appropriate Fixed MTRRs override the default access destination. Each fixed MTRR includes two bits, RdDram and WrDram, that determine the destination based on the access type. Writing reserved MemType values causes an error-on-write. If Core::X86::Msr::SYS\_CFG[SecureNestedPagingEn] is set to 1, this register will be read-only and attempts to write to this register will result in #GP(0).

\_lthree[1:0]\_core[7:0]\_nSIZE64K; MSR0000\_0250

Bits	Description																
63:61	Reserved.																
60	<b>RdDram_64K_70000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
59	<b>WrDram_64K_70000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
58:56	<b>MemType_64K_70000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
55:53	Reserved.																
52	<b>RdDram_64K_60000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
51	<b>WrDram_64K_60000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
50:48	<b>MemType_64K_60000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
47:45	Reserved.																
44	<b>RdDram_64K_50000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.																

	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
43	<b>WrDram_64K_50000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
42:40	<b>MemType_64K_50000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
39:37	Reserved.																
36	<b>RdDram_64K_40000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
35	<b>WrDram_64K_40000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
34:32	<b>MemType_64K_40000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
31:29	Reserved.																
28	<b>RdDram_64K_30000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
27	<b>WrDram_64K_30000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
26:24	<b>MemType_64K_30000: memory type.</b> Read-write. Reset: XXXb.																

	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
23:21	Reserved.	
20	<b>RdDram_64K_20000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
19	<b>WrDram_64K_20000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
18:16	<b>MemType_64K_20000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
15:13	Reserved.	
12	<b>RdDram_64K_10000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
11	<b>WrDram_64K_10000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
10:8	<b>MemType_64K_10000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
7:5	Reserved.	

4	<b>RdDram_64K_00000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. Address range from 00000h to 0FFFFh. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
3	<b>WrDram_64K_00000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. Address range from 00000h to 0FFFFh. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
2:0	<b>MemType_64K_00000: memory type.</b> Read-write. Reset: XXXb. Address range from 00000h to 0FFFFh.																
	<b>ValidValues:</b>																
	<table><tr><th>Value</th><th>Description</th></tr><tr><td>0h</td><td>UC or uncacheable.</td></tr><tr><td>1h</td><td>WC or write combining.</td></tr><tr><td>3h-2h</td><td>Reserved.</td></tr><tr><td>4h</td><td>WT or write through.</td></tr><tr><td>5h</td><td>WP or write protect.</td></tr><tr><td>6h</td><td>WB or write back.</td></tr><tr><td>7h</td><td>Reserved.</td></tr></table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
	Value	Description															
	0h	UC or uncacheable.															
	1h	WC or write combining.															
	3h-2h	Reserved.															
	4h	WT or write through.															
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																



**MSR0000\_0258 [Fixed-Size MTRRs] (Core::X86::Msrr::MtrrFix\_16K\_0)**

See Core::X86::Msrr::MtrrVarBase for general MTRR information. Fixed MTRRs are enabled through Core::X86::Msrr::MTRRdefType[MtrrDefTypeFixEn,MtrrDefTypeEn]. For addresses below 1MB, the appropriate Fixed MTRRs override the default access destination. Each fixed MTRR includes two bits, RdDram and WrDram, that determine the destination based on the access type. Writing reserved MemType values causes an error-on-write. If Core::X86::Msrr::SYS\_CFG[SecureNestedPagingEn] is set to 1, this register will be read-only and attempts to write to this register will result in #GP(0).

\_lthree[1:0]\_core[7:0]\_nSIZE16K0; MSR0000\_0258

Bits	Description																
63:61	Reserved.																
60	<b>RdDram_16K_9C000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
59	<b>WrDram_16K_9C000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
58:56	<b>MemType_16K_9C000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
55:53	Reserved.																
52	<b>RdDram_16K_98000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
51	<b>WrDram_16K_98000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
50:48	<b>MemType_16K_98000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
47:45	Reserved.																
44	<b>RdDram_16K_94000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.																



	AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
43	<b>WrDram_16K_94000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
42:40	<b>MemType_16K_94000: memory type.</b> Read-write. Reset: XXXb.																
	<b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
39:37	Reserved.																
36	<b>RdDram_16K_90000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
35	<b>WrDram_16K_90000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
34:32	<b>MemType_16K_90000: memory type.</b> Read-write. Reset: XXXb.																
	<b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
31:29	Reserved.																
28	<b>RdDram_16K_8C000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
27	<b>WrDram_16K_8C000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
26:24	<b>MemType_16K_8C000: memory type.</b> Read-write. Reset: XXXb.																

	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
23:21	Reserved.	
20	<b>RdDram_16K_88000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
19	<b>WrDram_16K_88000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
18:16	<b>MemType_16K_88000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
15:13	Reserved.	
12	<b>RdDram_16K_84000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
11	<b>WrDram_16K_84000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
10:8	<b>MemType_16K_84000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
7:5	Reserved.	

4	<b>RdDram_16K_80000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. Address range from 80000h to 83FFFh. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
3	<b>WrDram_16K_80000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. Address range from 80000h to 83FFFh. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
2:0	<b>MemType_16K_80000: memory type.</b> Read-write. Reset: XXXb. Address range from 80000h to 83FFFh.																
	<b>ValidValues:</b>																
	<table><tr><th>Value</th><th>Description</th></tr><tr><td>0h</td><td>UC or uncacheable.</td></tr><tr><td>1h</td><td>WC or write combining.</td></tr><tr><td>3h-2h</td><td>Reserved.</td></tr><tr><td>4h</td><td>WT or write through.</td></tr><tr><td>5h</td><td>WP or write protect.</td></tr><tr><td>6h</td><td>WB or write back.</td></tr><tr><td>7h</td><td>Reserved.</td></tr></table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
	Value	Description															
	0h	UC or uncacheable.															
	1h	WC or write combining.															
	3h-2h	Reserved.															
	4h	WT or write through.															
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																

**MSR0000\_0259 [Fixed-Size MTRRs] (Core::X86::Msrr::MtrrFix\_16K\_1)**

See Core::X86::Msrr::MtrrVarBase for general MTRR information. Fixed MTRRs are enabled through Core::X86::Msrr::MTRRdefType[MtrrDefTypeFixEn,MtrrDefTypeEn]. For addresses below 1MB, the appropriate Fixed MTRRs override the default access destination. Each fixed MTRR includes two bits, RdDram and WrDram, that determine the destination based on the access type. Writing reserved MemType values causes an error-on-write. If Core::X86::Msrr::SYS\_CFG[SecureNestedPagingEn] is set to 1, this register will be read-only and attempts to write to this register will result in #GP(0).

\_lthree[1:0]\_core[7:0]\_nSIZE16K1; MSR0000\_0259

Bits	Description																
63:61	Reserved.																
60	<b>RdDram_16K_BC000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
59	<b>WrDram_16K_BC000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
58:56	<b>MemType_16K_BC000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
55:53	Reserved.																
52	<b>RdDram_16K_B8000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
51	<b>WrDram_16K_B8000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
50:48	<b>MemType_16K_B8000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
47:45	Reserved.																
44	<b>RdDram_16K_B4000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.																

	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
43	<b>WrDram_16K_B4000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
42:40	<b>MemType_16K_B4000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
39:37	Reserved.																
36	<b>RdDram_16K_B0000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
35	<b>WrDram_16K_B0000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
34:32	<b>MemType_16K_B0000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
31:29	Reserved.																
28	<b>RdDram_16K_AC000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
27	<b>WrDram_16K_AC000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
26:24	<b>MemType_16K_AC000: memory type.</b> Read-write. Reset: XXXb.																

	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
23:21	Reserved.	
20	<b>RdDram_16K_A8000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
19	<b>WrDram_16K_A8000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
18:16	<b>MemType_16K_A8000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
15:13	Reserved.	
12	<b>RdDram_16K_A4000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
11	<b>WrDram_16K_A4000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
10:8	<b>MemType_16K_A4000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
7:5	Reserved.	

4	<b>RdDram_16K_A0000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. Address range from A0000h to A3FFFh. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks Reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
3	<b>WrDram_16K_A0000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. Address range from A0000h to A3FFFh. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks Reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
2:0	<b>MemType_16K_A0000: memory type.</b> Read-write. Reset: XXXb. Address range from A0000h to A3FFFh.																
	<b>ValidValues:</b>																
	<table><tr><th>Value</th><th>Description</th></tr><tr><td>0h</td><td>UC or uncacheable.</td></tr><tr><td>1h</td><td>WC or write combining.</td></tr><tr><td>3h-2h</td><td>Reserved.</td></tr><tr><td>4h</td><td>WT or write through.</td></tr><tr><td>5h</td><td>WP or write protect.</td></tr><tr><td>6h</td><td>WB or write back.</td></tr><tr><td>7h</td><td>Reserved.</td></tr></table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
	Value	Description															
	0h	UC or uncacheable.															
	1h	WC or write combining.															
	3h-2h	Reserved.															
	4h	WT or write through.															
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																

**MSR0000\_0268 [Fixed-Size MTRRs] (Core::X86::Msrr::MtrrFix\_4K\_0)**

See Core::X86::Msrr::MtrrVarBase for general MTRR information. Fixed MTRRs are enabled through Core::X86::Msrr::MTRRdefType[MtrrDefTypeFixEn,MtrrDefTypeEn]. For addresses below 1MB, the appropriate Fixed MTRRs override the default access destination. Each fixed MTRR includes two bits, RdDram and WrDram, that determine the destination based on the access type. Writing Reserved MemType values causes an error-on-write. If Core::X86::Msrr::SYS\_CFG[SecureNestedPagingEn] is set to 1, this register will be read-only and attempts to write to this register will result in #GP(0).

\_lthree[1:0]\_core[7:0]\_nSIZE4K0; MSR0000\_0268

Bits	Description																
63:61	Reserved.																
60	<b>RdDram_4K_C7000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
59	<b>WrDram_4K_C7000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
58:56	<b>MemType_4K_C7000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
55:53	Reserved.																
52	<b>RdDram_4K_C6000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
51	<b>WrDram_4K_C6000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
50:48	<b>MemType_4K_C6000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
47:45	Reserved.																
44	<b>RdDram_4K_C5000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.																



	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
43	<b>WrDram_4K_C5000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
42:40	<b>MemType_4K_C5000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
39:37	Reserved.																
36	<b>RdDram_4K_C4000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
35	<b>WrDram_4K_C4000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
34:32	<b>MemType_4K_C4000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
31:29	Reserved.																
28	<b>RdDram_4K_C3000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
27	<b>WrDram_4K_C3000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
26:24	<b>MemType_4K_C3000: memory type.</b> Read-write. Reset: XXXb.																

	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
23:21	Reserved.	
20	<b>RdDram_4K_C2000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
19	<b>WrDram_4K_C2000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
18:16	<b>MemType_4K_C2000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
15:13	Reserved.	
12	<b>RdDram_4K_C1000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
11	<b>WrDram_4K_C1000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
10:8	<b>MemType_4K_C1000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
7:5	Reserved.	

4	<b>RdDram_4K_C0000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. Address range from C0000h to C0FFFh. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks Reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
3	<b>WrDram_4K_C0000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. Address range from C0000h to C0FFFh. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks Reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
2:0	<b>MemType_4K_C0000: memory type.</b> Read-write. Reset: XXXb. Address range from C0000h to C0FFFh.																
	<b>ValidValues:</b>																
	<table><tr><th>Value</th><th>Description</th></tr><tr><td>0h</td><td>UC or uncacheable.</td></tr><tr><td>1h</td><td>WC or write combining.</td></tr><tr><td>3h-2h</td><td>Reserved.</td></tr><tr><td>4h</td><td>WT or write through.</td></tr><tr><td>5h</td><td>WP or write protect.</td></tr><tr><td>6h</td><td>WB or write back.</td></tr><tr><td>7h</td><td>Reserved.</td></tr></table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
	Value	Description															
	0h	UC or uncacheable.															
	1h	WC or write combining.															
	3h-2h	Reserved.															
	4h	WT or write through.															
	5h	WP or write protect.															
6h	WB or write back.																
7h	Reserved.																

**MSR0000\_0269 [Fixed-Size MTRRs] (Core::X86::Msrr::MtrrFix\_4K\_1)**

See Core::X86::Msrr::MtrrVarBase for general MTRR information. Fixed MTRRs are enabled through Core::X86::Msrr::MTRRdefType[MtrrDefTypeFixEn,MtrrDefTypeEn]. For addresses below 1MB, the appropriate Fixed MTRRs override the default access destination. Each fixed MTRR includes two bits, RdDram and WrDram, that determine the destination based on the access type. Writing Reserved MemType values causes an error-on-write. If Core::X86::Msrr::SYS\_CFG[SecureNestedPagingEn] is set to 1, this register will be read-only and attempts to write to this register will result in #GP(0).

\_lthree[1:0]\_core[7:0]\_nSIZE4K1; MSR0000\_0269

Bits	Description																
63:61	Reserved.																
60	<b>RdDram_4K_CF000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
59	<b>WrDram_4K_CF000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
58:56	<b>MemType_4K_CF000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
55:53	Reserved.																
52	<b>RdDram_4K_CE000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
51	<b>WrDram_4K_CE000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
50:48	<b>MemType_4K_CE000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
47:45	Reserved.																
44	<b>RdDram_4K_CD000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.																

	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
43	<b>WrDram_4K_CD000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
42:40	<b>MemType_4K_CD000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
39:37	Reserved.																
36	<b>RdDram_4K_CC000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
35	<b>WrDram_4K_CC000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
34:32	<b>MemType_4K_CC000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
31:29	Reserved.																
28	<b>RdDram_4K_CB000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
27	<b>WrDram_4K_CB000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
26:24	<b>MemType_4K_CB000: memory type.</b> Read-write. Reset: XXXb.																

	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
23:21	Reserved.	
20	<b>RdDram_4K_CA000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
19	<b>WrDram_4K_CA000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
18:16	<b>MemType_4K_CA000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
15:13	Reserved.	
12	<b>RdDram_4K_C9000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
11	<b>WrDram_4K_C9000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
10:8	<b>MemType_4K_C9000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
7:5	Reserved.	

4	<b>RdDram_4K_C8000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. Address range from C8000 to C8FFF. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks Reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
3	<b>WrDram_4K_C8000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. Address range from C8000 to C8FFF. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks Reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
2:0	<b>MemType_4K_C8000: memory type.</b> Read-write. Reset: XXXb. Address range from C8000 to C8FFF.																
	<b>ValidValues:</b>																
	<table><tr><th>Value</th><th>Description</th></tr><tr><td>0h</td><td>UC or uncacheable.</td></tr><tr><td>1h</td><td>WC or write combining.</td></tr><tr><td>3h-2h</td><td>Reserved.</td></tr><tr><td>4h</td><td>WT or write through.</td></tr><tr><td>5h</td><td>WP or write protect.</td></tr><tr><td>6h</td><td>WB or write back.</td></tr><tr><td>7h</td><td>Reserved.</td></tr></table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
	Value	Description															
	0h	UC or uncacheable.															
	1h	WC or write combining.															
	3h-2h	Reserved.															
	4h	WT or write through.															
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																

**MSR0000\_026A [Fixed-Size MTRRs] (Core::X86::Msrr::MtrrFix\_4K\_2)**

See Core::X86::Msrr::MtrrVarBase for general MTRR information. Fixed MTRRs are enabled through Core::X86::Msrr::MTRRdefType[MtrrDefTypeFixEn,MtrrDefTypeEn]. For addresses below 1MB, the appropriate Fixed MTRRs override the default access destination. Each fixed MTRR includes two bits, RdDram and WrDram, that determine the destination based on the access type. Writing Reserved MemType values causes an error-on-write. If Core::X86::Msrr::SYS\_CFG[SecureNestedPagingEn] is set to 1, this register will be read-only and attempts to write to this register will result in #GP(0).

\_lthree[1:0]\_core[7:0]\_nSIZE4K2; MSR0000\_026A

Bits	Description																
63:61	Reserved.																
60	<b>RdDram_4K_D7000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
59	<b>WrDram_4K_D7000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
58:56	<b>MemType_4K_D7000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
55:53	Reserved.																
52	<b>RdDram_4K_D6000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
51	<b>WrDram_4K_D6000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
50:48	<b>MemType_4K_D6000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
47:45	Reserved.																
44	<b>RdDram_4K_D5000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.																



	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
43	<b>WrDram_4K_D5000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
42:40	<b>MemType_4K_D5000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
39:37	Reserved.																
36	<b>RdDram_4K_D4000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
35	<b>WrDram_4K_D4000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
34:32	<b>MemType_4K_D4000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
31:29	Reserved.																
28	<b>RdDram_4K_D3000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
27	<b>WrDram_4K_D3000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
26:24	<b>MemType_4K_D3000: memory type.</b> Read-write. Reset: XXXb.																

	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
23:21	Reserved.	
20	<b>RdDram_4K_D2000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
19	<b>WrDram_4K_D2000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
18:16	<b>MemType_4K_D2000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
15:13	Reserved.	
12	<b>RdDram_4K_D1000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
11	<b>WrDram_4K_D1000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
10:8	<b>MemType_4K_D1000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
7:5	Reserved.	

4	<b>RdDram_4K_D0000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. Address range from D0000h to D0FFFh. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks Reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
3	<b>WrDram_4K_D0000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. Address range from D0000h to D0FFFh. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
2:0	<b>MemType_4K_D0000: memory type.</b> Read-write. Reset: XXXb. Address range from D0000h to D0FFFh.																
	<b>ValidValues:</b>																
	<table><tr><th>Value</th><th>Description</th></tr><tr><td>0h</td><td>UC or uncacheable.</td></tr><tr><td>1h</td><td>WC or write combining.</td></tr><tr><td>3h-2h</td><td>Reserved.</td></tr><tr><td>4h</td><td>WT or write through.</td></tr><tr><td>5h</td><td>WP or write protect.</td></tr><tr><td>6h</td><td>WB or write back.</td></tr><tr><td>7h</td><td>Reserved.</td></tr></table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
	Value	Description															
	0h	UC or uncacheable.															
	1h	WC or write combining.															
	3h-2h	Reserved.															
	4h	WT or write through.															
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																

**MSR0000\_026B [Fixed-Size MTRRs] (Core::X86::Msrr::MtrrFix\_4K\_3)**

See Core::X86::Msrr::MtrrVarBase for general MTRR information. Fixed MTRRs are enabled through Core::X86::Msrr::MTRRdefType[MtrrDefTypeFixEn,MtrrDefTypeEn]. For addresses below 1MB, the appropriate Fixed MTRRs override the default access destination. Each fixed MTRR includes two bits, RdDram and WrDram, that determine the destination based on the access type. Writing Reserved MemType values causes an error-on-write. If Core::X86::Msrr::SYS\_CFG[SecureNestedPagingEn] is set to 1, this register will be read-only and attempts to write to this register will result in #GP(0).

\_lthree[1:0]\_core[7:0]\_nSIZE4K3; MSR0000\_026B

Bits	Description																
63:61	Reserved.																
60	<b>RdDram_4K_DF000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
59	<b>WrDram_4K_DF000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
58:56	<b>MemType_4K_DF000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
55:53	Reserved.																
52	<b>RdDram_4K_DE000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
51	<b>WrDram_4K_DE000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
50:48	<b>MemType_4K_DE000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
47:45	Reserved.																
44	<b>RdDram_4K_DD000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.																

	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
43	<b>WrDram_4K_DD000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
42:40	<b>MemType_4K_DD000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
39:37	Reserved.																
36	<b>RdDram_4K_DC000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
35	<b>WrDram_4K_DC000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
34:32	<b>MemType_4K_DC000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
31:29	Reserved.																
28	<b>RdDram_4K_DB000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
27	<b>WrDram_4K_DB000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
26:24	<b>MemType_4K_DB000: memory type.</b> Read-write. Reset: XXXb.																

	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
23:21	Reserved.	
20	<b>RdDram_4K_DA000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
19	<b>WrDram_4K_DA000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
18:16	<b>MemType_4K_DA000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
15:13	Reserved.	
12	<b>RdDram_4K_D9000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
11	<b>WrDram_4K_D9000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
10:8	<b>MemType_4K_D9000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
7:5	Reserved.	

4	<b>RdDram_4K_D8000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. Address range from D8000h to D8FFFh. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks Reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
3	<b>WrDram_4K_D8000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. Address range from D8000h to D8FFFh. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks Reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
2:0	<b>MemType_4K_D8000: memory type.</b> Read-write. Reset: XXXb. Address range from D8000h to D8FFFh.																
	<b>ValidValues:</b>																
	<table><tr><th>Value</th><th>Description</th></tr><tr><td>0h</td><td>UC or uncacheable.</td></tr><tr><td>1h</td><td>WC or write combining.</td></tr><tr><td>3h-2h</td><td>Reserved.</td></tr><tr><td>4h</td><td>WT or write through.</td></tr><tr><td>5h</td><td>WP or write protect.</td></tr><tr><td>6h</td><td>WB or write back.</td></tr><tr><td>7h</td><td>Reserved.</td></tr></table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
	Value	Description															
	0h	UC or uncacheable.															
	1h	WC or write combining.															
	3h-2h	Reserved.															
	4h	WT or write through.															
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																

**MSR0000\_026C [Fixed-Size MTRRs] (Core::X86::Msr::MtrrFix\_4K\_4)**

See Core::X86::Msr::MtrrVarBase for general MTRR information. Fixed MTRRs are enabled through Core::X86::Msr::MTRRdefType[MtrrDefTypeFixEn,MtrrDefTypeEn]. For addresses below 1MB, the appropriate Fixed MTRRs override the default access destination. Each fixed MTRR includes two bits, RdDram and WrDram, that determine the destination based on the access type. Writing reserved MemType values causes an error-on-write. If Core::X86::Msr::SYS\_CFG[SecureNestedPagingEn] is set to 1, this register will be read-only and attempts to write to this register will result in #GP(0).

\_lthree[1:0]\_core[7:0]\_nSIZE4K4; MSR0000\_026C

Bits	Description																
63:61	Reserved.																
60	<b>RdDram_4K_E7000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
59	<b>WrDram_4K_E7000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
58:56	<b>MemType_4K_E7000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
55:53	Reserved.																
52	<b>RdDram_4K_E6000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
51	<b>WrDram_4K_E6000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
50:48	<b>MemType_4K_E6000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
47:45	Reserved.																
44	<b>RdDram_4K_E5000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.																



	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
43	<b>WrDram_4K_E5000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
42:40	<b>MemType_4K_E5000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
39:37	Reserved.																
36	<b>RdDram_4K_E4000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
35	<b>WrDram_4K_E4000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
34:32	<b>MemType_4K_E4000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
31:29	Reserved.																
28	<b>RdDram_4K_E3000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
27	<b>WrDram_4K_E3000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
26:24	<b>MemType_4K_E3000: memory type.</b> Read-write. Reset: XXXb.																

	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
23:21	Reserved.	
20	<b>RdDram_4K_E2000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
19	<b>WrDram_4K_E2000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
18:16	<b>MemType_4K_E2000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
15:13	Reserved.	
12	<b>RdDram_4K_E1000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
11	<b>WrDram_4K_E1000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
10:8	<b>MemType_4K_E1000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
7:5	Reserved.	

4	<b>RdDram_4K_E0000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. Address range from E0000h to E0FFFh. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks Reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
3	<b>WrDram_4K_E0000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. Address range from E0000h to E0FFFh. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks Reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
2:0	<b>MemType_4K_E0000: memory type.</b> Read-write. Reset: XXXb. Address range from E0000h to E0FFFh.																
	<b>ValidValues:</b>																
	<table><tr><th>Value</th><th>Description</th></tr><tr><td>0h</td><td>UC or uncacheable.</td></tr><tr><td>1h</td><td>WC or write combining.</td></tr><tr><td>3h-2h</td><td>Reserved.</td></tr><tr><td>4h</td><td>WT or write through.</td></tr><tr><td>5h</td><td>WP or write protect.</td></tr><tr><td>6h</td><td>WB or write back.</td></tr><tr><td>7h</td><td>Reserved.</td></tr></table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
	Value	Description															
	0h	UC or uncacheable.															
	1h	WC or write combining.															
	3h-2h	Reserved.															
	4h	WT or write through.															
	5h	WP or write protect.															
6h	WB or write back.																
7h	Reserved.																

**MSR0000\_026D [Fixed-Size MTRRs] (Core::X86::Msrr::MtrrFix\_4K\_5)**

See Core::X86::Msrr::MtrrVarBase for general MTRR information. Fixed MTRRs are enabled through Core::X86::Msrr::MTRRdefType[MtrrDefTypeFixEn,MtrrDefTypeEn]. For addresses below 1MB, the appropriate Fixed MTRRs override the default access destination. Each fixed MTRR includes two bits, RdDram and WrDram, that determine the destination based on the access type. Writing Reserved MemType values causes an error-on-write. If Core::X86::Msrr::SYS\_CFG[SecureNestedPagingEn] is set to 1, this register will be read-only and attempts to write to this register will result in #GP(0).

\_lthree[1:0]\_core[7:0]\_nSIZE4K5; MSR0000\_026D

Bits	Description																
63:61	Reserved.																
60	<b>RdDram_4K_EF000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
59	<b>WrDram_4K_EF000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
58:56	<b>MemType_4K_EF000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
55:53	Reserved.																
52	<b>RdDram_4K_EE000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
51	<b>WrDram_4K_EE000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
50:48	<b>MemType_4K_EE000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
47:45	Reserved.																
44	<b>RdDram_4K_ED000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.																

	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
43	<b>WrDram_4K_ED000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
42:40	<b>MemType_4K_ED000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
39:37	Reserved.																
36	<b>RdDram_4K_EC000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
35	<b>WrDram_4K_EC000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
34:32	<b>MemType_4K_EC000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
31:29	Reserved.																
28	<b>RdDram_4K_EB000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
27	<b>WrDram_4K_EB000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
26:24	<b>MemType_4K_EB000: memory type.</b> Read-write. Reset: XXXb.																

	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
23:21	Reserved.	
20	<b>RdDram_4K_EA000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
19	<b>WrDram_4K_EA000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
18:16	<b>MemType_4K_EA000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
15:13	Reserved.	
12	<b>RdDram_4K_E9000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
11	<b>WrDram_4K_E9000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
10:8	<b>MemType_4K_E9000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
7:5	Reserved.	

4	<b>RdDram_4K_E8000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. Address range from E8000h to E8FFFh. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks Reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
3	<b>WrDram_4K_E8000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. Address range from E8000h to E8FFFh. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks Reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
2:0	<b>MemType_4K_E8000: memory type.</b> Read-write. Reset: XXXb. Address range from E8000h to E8FFFh.																
	<b>ValidValues:</b>																
	<table><tr><th>Value</th><th>Description</th></tr><tr><td>0h</td><td>UC or uncacheable.</td></tr><tr><td>1h</td><td>WC or write combining.</td></tr><tr><td>3h-2h</td><td>Reserved.</td></tr><tr><td>4h</td><td>WT or write through.</td></tr><tr><td>5h</td><td>WP or write protect.</td></tr><tr><td>6h</td><td>WB or write back.</td></tr><tr><td>7h</td><td>Reserved.</td></tr></table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
	Value	Description															
	0h	UC or uncacheable.															
	1h	WC or write combining.															
	3h-2h	Reserved.															
	4h	WT or write through.															
	5h	WP or write protect.															
6h	WB or write back.																
7h	Reserved.																

**MSR0000\_026E [Fixed-Size MTRRs] (Core::X86::Msrr::MtrrFix\_4K\_6)**

See Core::X86::Msrr::MtrrVarBase for general MTRR information. Fixed MTRRs are enabled through Core::X86::Msrr::MTRRdefType[MtrrDefTypeFixEn,MtrrDefTypeEn]. For addresses below 1MB, the appropriate Fixed MTRRs override the default access destination. Each fixed MTRR includes two bits, RdDram and WrDram, that determine the destination based on the access type. Writing Reserved MemType values causes an error-on-write. If Core::X86::Msrr::SYS\_CFG[SecureNestedPagingEn] is set to 1, this register will be read-only and attempts to write to this register will result in #GP(0).

\_lthree[1:0]\_core[7:0]\_nSIZE4K6; MSR0000\_026E

Bits	Description																
63:61	Reserved.																
60	<b>RdDram_4K_F7000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
59	<b>WrDram_4K_F7000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
58:56	<b>MemType_4K_F7000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
55:53	Reserved.																
52	<b>RdDram_4K_F6000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
51	<b>WrDram_4K_F6000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
50:48	<b>MemType_4K_F6000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
47:45	Reserved.																
44	<b>RdDram_4K_F5000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.																



	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
43	<b>WrDram_4K_F5000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
42:40	<b>MemType_4K_F5000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
39:37	Reserved.																
36	<b>RdDram_4K_F4000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
35	<b>WrDram_4K_F4000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
34:32	<b>MemType_4K_F4000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
31:29	Reserved.																
28	<b>RdDram_4K_F3000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
27	<b>WrDram_4K_F3000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
26:24	<b>MemType_4K_F3000: memory type.</b> Read-write. Reset: XXXb.																

	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
23:21	Reserved.	
20	<b>RdDram_4K_F2000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
19	<b>WrDram_4K_F2000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
18:16	<b>MemType_4K_F2000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
15:13	Reserved.	
12	<b>RdDram_4K_F1000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
11	<b>WrDram_4K_F1000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
10:8	<b>MemType_4K_F1000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
7:5	Reserved.	

4	<b>RdDram_4K_F0000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. Address range from F0000h to F0FFF. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks Reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
3	<b>WrDram_4K_F0000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. Address range from F0000h to F0FFF. Core::X86::Msr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks Reads of the stored value.																
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
2:0	<b>MemType_4K_F0000: memory type.</b> Read-write. Reset: XXXb. Address range from F0000h to F0FFFh.																
	<b>ValidValues:</b>																
	<table><tr><th>Value</th><th>Description</th></tr><tr><td>0h</td><td>UC or uncacheable.</td></tr><tr><td>1h</td><td>WC or write combining.</td></tr><tr><td>3h-2h</td><td>Reserved.</td></tr><tr><td>4h</td><td>WT or write through.</td></tr><tr><td>5h</td><td>WP or write protect.</td></tr><tr><td>6h</td><td>WB or write back.</td></tr><tr><td>7h</td><td>Reserved.</td></tr></table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
	Value	Description															
	0h	UC or uncacheable.															
	1h	WC or write combining.															
	3h-2h	Reserved.															
	4h	WT or write through.															
	5h	WP or write protect.															
6h	WB or write back.																
7h	Reserved.																

**MSR0000\_026F [Fixed-Size MTRRs] (Core::X86::Msrr::MtrrFix\_4K\_7)**

See Core::X86::Msrr::MtrrVarBase for general MTRR information. Fixed MTRRs are enabled through Core::X86::Msrr::MTRRdefType[MtrrDefTypeFixEn,MtrrDefTypeEn]. For addresses below 1MB, the appropriate Fixed MTRRs override the default access destination. Each fixed MTRR includes two bits, RdDram and WrDram, that determine the destination based on the access type. Writing Reserved MemType values causes an error-on-write. If Core::X86::Msrr::SYS\_CFG[SecureNestedPagingEn] is set to 1, this register will be read-only and attempts to write to this register will result in #GP(0).

\_lthree[1:0]\_core[7:0]\_nSIZE4K7; MSR0000\_026F

Bits	Description																
63:61	Reserved.																
60	<b>RdDram_4K_FF000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
59	<b>WrDram_4K_FF000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
58:56	<b>MemType_4K_FF000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
55:53	Reserved.																
52	<b>RdDram_4K_FE000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
51	<b>WrDram_4K_FE000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
50:48	<b>MemType_4K_FE000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
47:45	Reserved.																
44	<b>RdDram_4K_FD000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.																

	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
43	<b>WrDram_4K_FD000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
42:40	<b>MemType_4K_FD000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
39:37	Reserved.																
36	<b>RdDram_4K_FC000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
35	<b>WrDram_4K_FC000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
34:32	<b>MemType_4K_FC000: memory type.</b> Read-write. Reset: XXXb. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
31:29	Reserved.																
28	<b>RdDram_4K_FB000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
27	<b>WrDram_4K_FB000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
26:24	<b>MemType_4K_FB000: memory type.</b> Read-write. Reset: XXXb.																

	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
23:21	Reserved.	
20	<b>RdDram_4K_FA000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
19	<b>WrDram_4K_FA000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
18:16	<b>MemType_4K_FA000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
15:13	Reserved.	
12	<b>RdDram_4K_F9000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
11	<b>WrDram_4K_F9000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM.	
	AccessType: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.	
10:8	<b>MemType_4K_F9000: memory type.</b> Read-write. Reset: XXXb.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	UC or uncacheable.
	1h	WC or write combining.
	3h-2h	Reserved.
	4h	WT or write through.
	5h	WP or write protect.
	6h	WB or write back.
	7h	Reserved.
7:5	Reserved.	

4	<b>RdDram_4K_F8000: read DRAM.</b> 0=Read accesses to the range are marked as MMIO. 1=Read accesses to the range are marked as destined for DRAM. Address range from F8000h to F8FFFh. Core::X86::Msrr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks Reads of the stored value.																
	AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
3	<b>WrDram_4K_F8000: write DRAM.</b> 0=Write accesses to the range are marked as MMIO. 1=Write accesses to the range are marked as destined for DRAM. Address range from F8000h to F8FFFh. Core::X86::Msrr::SYS_CFG[MtrrFixDramEn,MtrrFixDramModEn] masks Reads of the stored value.																
	AccessType: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? Read-write : Read,Error-on-write-1. Reset: Core::X86::Msrr::SYS_CFG[MtrrFixDramModEn] ? X : Fixed,0.																
2:0	<b>MemType_4K_F8000: memory type.</b> Read-write. Reset: XXXb. Address range from F8000h to F8FFFh.																
	<b>ValidValues:</b>																
	<table><tr><th>Value</th><th>Description</th></tr><tr><td>0h</td><td>UC or uncacheable.</td></tr><tr><td>1h</td><td>WC or write combining.</td></tr><tr><td>3h-2h</td><td>Reserved.</td></tr><tr><td>4h</td><td>WT or write through.</td></tr><tr><td>5h</td><td>WP or write protect.</td></tr><tr><td>6h</td><td>WB or write back.</td></tr><tr><td>7h</td><td>Reserved.</td></tr></table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
	Value	Description															
	0h	UC or uncacheable.															
	1h	WC or write combining.															
	3h-2h	Reserved.															
	4h	WT or write through.															
	5h	WP or write protect.															
6h	WB or write back.																
7h	Reserved.																

**MSR0000\_0277 [Page Attribute Table] (Core::X86::Msr::PAT)**

This register specifies the memory type based on the PAT, PCD, and PWT bits in the virtual address page tables.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0277

Bits	Description																
63:59	Reserved.																
58:56	<b>PA7MemType.</b> Read-write. Reset: 0h. Default UC. MemType for {PAT, PCD, PWT} = 7h.																
	<b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
55:51	Reserved.																
50:48	<b>PA6MemType.</b> Read-write. Reset: 7h. Default UC. MemType for {PAT, PCD, PWT} = 6h.																
	<b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>UC minus.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	UC minus.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	UC minus.																
47:43	Reserved.																
42:40	<b>PA5MemType.</b> Read-write. Reset: 4h. Default WT. MemType for {PAT, PCD, PWT} = 5h.																
	<b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
39:35	Reserved.																
34:32	<b>PA4MemType.</b> Read-write. Reset: 6h. Default WB. MemType for {PAT, PCD, PWT} = 4h.																
	<b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																



31:27	Reserved.																
26:24	<b>PA3MemType.</b> Read-write. Reset: 0h. Default UC. MemType for {PAT, PCD, PWT} = 3h.																
	<b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
23:19	Reserved.																
18:16	<b>PA2MemType.</b> Read-write. Reset: 7h. Default UC. MemType for {PAT, PCD, PWT} = 2h.																
	<b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>UC minus.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	UC minus.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	UC minus.																
15:11	Reserved.																
10:8	<b>PA1MemType.</b> Read-write. Reset: 4h. Default WT. MemType for {PAT, PCD, PWT} = 1h.																
	<b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																
7:3	Reserved.																
2:0	<b>PA0MemType.</b> Read-write. Reset: 6h. MemType for {PAT, PCD, PWT} = 0h.																
	<b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>UC or uncacheable.</td></tr> <tr> <td>1h</td><td>WC or write combining.</td></tr> <tr> <td>3h-2h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>WT or write through.</td></tr> <tr> <td>5h</td><td>WP or write protect.</td></tr> <tr> <td>6h</td><td>WB or write back.</td></tr> <tr> <td>7h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	UC or uncacheable.	1h	WC or write combining.	3h-2h	Reserved.	4h	WT or write through.	5h	WP or write protect.	6h	WB or write back.	7h	Reserved.
Value	Description																
0h	UC or uncacheable.																
1h	WC or write combining.																
3h-2h	Reserved.																
4h	WT or write through.																
5h	WP or write protect.																
6h	WB or write back.																
7h	Reserved.																

**MSR0000\_02FF [MTRR Default Memory Type] (Core::X86::Msrr::MTRRdefType)**

See Core::X86::Msrr::MtrrVarBase for general MTRR information.

\_lthree[1:0]\_core[7:0]; MSR0000\_02FF

Bits	Description
63:12	Reserved.
11	<b>MtrrDefTypeEn: variable and fixed MTRR enable.</b> Read-write. Reset: 0. 0=Fixed and variable MTRRs are not enabled. 1=Core::X86::Msrr::MtrrVarBase, and Core::X86::Msrr::MtrrFix_64K through Core::X86::Msrr::MtrrFix_4K_7 are enabled.
10	<b>MtrrDefTypeFixEn: fixed MTRR enable.</b> Read-write. Reset: 0. 0=Core::X86::Msrr::MtrrFix_64K through Core::X86::Msrr::MtrrFix_4K_7 are not enabled. 1=Core::X86::Msrr::MtrrFix_64K through Core::X86::Msrr::MtrrFix_4K_7 are enabled. This field is ignored (and the fixed MTRRs are not enabled) if Core::X86::Msrr::MTRRdefType[MtrrDefTypeEn] == 0.
9:8	Reserved.
7:0	<b>MemType: memory type.</b> Read-write. Reset: 00h. <b>Description:</b> If MtrrDefTypeEn == 1 then MemType specifies the memory type for memory space that is not specified by either the fixed or variable range MTRRs. If MtrrDefTypeEn == 0 then the default memory type for all of memory is UC. Valid encodings are {00000b, Core::X86::Msrr::MtrrFix_64K through Core::X86::Msrr::MtrrFix_4K_7[2:0]}. Other Write values cause a GP(0).

**MSR0000\_06A0 [User CET] (Core::X86::Msrr::U\_CET)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_06A0

Bits	Description
63:2	Reserved.
1	<b>WRSHSTKEN.</b> Read-write. Reset: 0. Enables the WRSS instruction in User Mode.
0	<b>SHSTKEN.</b> Read-write. Reset: 0. When Set Shadow stack is enabled in User mode.

**MSR0000\_06A2 [Supervisor CET] (Core::X86::Msrr::S\_CET)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_06A2

Bits	Description
63:2	Reserved.
1	<b>WRSHSTKEN.</b> Read-write. Reset: 0. Enables the WRSS instruction in Supervisor Mode.
0	<b>SHSTKEN.</b> Read-write. Reset: 0. When Set Shadow stack is enabled in Supervisor mode.

**MSR0000\_06A4 [PL0 Shadow Stack Pointer] (Core::X86::Msrr::PL0Ssp)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_06A4

Bits	Description
63:2	<b>UserLinAddress: PL0 user top of SSP.</b> Read-write. Reset: 0000_0000_0000_0000h. UserLinAddress[63:32] must be zero in 32-bit mode.
1:0	Reserved.

**MSR0000\_06A5 [PL1 Shadow Stack Pointer] (Core::X86::Msr::PL1Ssp)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_06A5

Bits	Description
63:2	<b>UserLinAddress: PL1 user top of SSP.</b> Read-write. Reset: 0000_0000_0000_0000h. UserLinAddress[63:32] must be zero in 32-bit mode.
1:0	Reserved.

**MSR0000\_06A6 [PL2 Shadow Stack Pointer] (Core::X86::Msr::PL2Ssp)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_06A6

Bits	Description
63:2	<b>UserLinAddress: PL2 user top of SSP.</b> Read-write. Reset: 0000_0000_0000_0000h. UserLinAddress[63:32] must be zero in 32-bit mode.
1:0	Reserved.

**MSR0000\_06A7 [PL3 Shadow Stack Pointer] (Core::X86::Msr::PL3Ssp)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_06A7

Bits	Description
63:2	<b>UserLinAddress: PL3 user top of SSP.</b> Read-write. Reset: 0000_0000_0000_0000h. UserLinAddress[63:32] must be zero in 32-bit mode.
1:0	Reserved.

**MSR0000\_06A8 [Interrupt SSP Table Address] (Core::X86::Msr::IstSspAddr)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_06A8

Bits	Description
63:0	<b>IntrLinTableAddress.</b> Read-write. Reset: 0000_0000_0000_0000h. Shadow Stack Pointer interrupt table.

**MSR0000\_0802 [APIC ID] (Core::X86::Msr::APIC\_ID)**

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0802

Bits	Description
63:32	Reserved.
31:0	<b>ApicId[31:0]: APIC ID[31:0].</b> Reset: XXXX_XXXXh. Local x2APIC ID register. AccessType: X2APICEN ? Read-only,Error-on-write : Error-on-read,Error-on-write.

**MSR0000\_0803 [APIC Version] (Core::X86::Msr::ApicVersion)**

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0803

Bits	Description
63:32	Reserved.
31	<b>ExtApicSpace: extended APIC register space present.</b> Reset: 1. 1=Indicates the presence of extended APIC register space starting at Core::X86::Msr::ExtendedApicFeature. AccessType: X2APICEN ? Read-only,Error-on-write : Error-on-read,Error-on-write.
30:25	Reserved.
24	<b>DirectedEoiSupport: directed EOI support.</b> Reset: 1. 0=Directed EOI capability not supported. 1=Directed EOI capability supported. AccessType: X2APICEN ? Read-only,Error-on-write : Error-on-read,Error-on-write.
23:16	<b>MaxLvtEntry.</b> Reset: XXh. Specifies the number of entries in the local vector table minus one. AccessType: X2APICEN ? Read-only,Error-on-write : Error-on-read,Error-on-write.
15:8	Reserved.
7:0	<b>Version.</b> Reset: 10h. Indicates the version number of this APIC implementation. AccessType: X2APICEN ? Read-only,Error-on-write : Error-on-read,Error-on-write.

**MSR0000\_0808 [Task Priority] (Core::X86::Msr::TPR)**

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0808

Bits	Description
63:8	Reserved.
7:0	<b>Priority.</b> Reset: 00h. This field is assigned by software to set a threshold priority at which the core is interrupted. AccessType: X2APICEN ? Read-write, Volatile : Error-on-read,Error-on-write.

**MSR0000\_0809 [Arbitration Priority] (Core::X86::Msr::ArbitrationPriority)**

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0809

Bits	Description
63:8	Reserved.
7:0	<b>Priority.</b> Reset: 00h. Indicates the current priority for a pending interrupt, or a task or interrupt being serviced by the core. The priority is used to arbitrate between cores to determine which accepts a lowest-priority interrupt request. AccessType: X2APICEN ? Read-only,Error-on-write, Volatile : Error-on-read,Error-on-write.

**MSR0000\_080A [Processor Priority] (Core::X86::Msr::ProcessorPriority)**

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_080A

Bits	Description
63:8	Reserved.
7:0	<b>Priority.</b> Reset: 00h. Indicates the core's current priority servicing a task or interrupt, and is used to determine if any pending interrupts should be serviced. It is the higher value of the task priority value and the current highest in-service interrupt. AccessType: X2APICEN ? Read-only,Error-on-write, Volatile : Error-on-read,Error-on-write.

**MSR0000\_080B [End Of Interrupt] (Core::X86::Msr::EOI)**

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_080B

Bits	Description
63:0	<b>EOI.</b> Reset: 0000_0000_0000_0000h. A Write zero to this field indicates the end of interrupt processing the currently in service interrupt. AccessType: X2APICEN ? Write-0-only,Error-on-read,Error-on-write-1 : Error-on-read,Error-on-write.

**MSR0000\_080D [Logical Destination Register] (Core::X86::Msr::LDR)**

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_080D

Bits	Description																																		
63:32	Reserved.																																		
31:16	<b>ClusterDestination.</b> Reset: 0000h. Specifies cluster's destination identification. AccessType: X2APICEN ? Read-only,Error-on-write : Error-on-read,Error-on-write.																																		
15:0	<b>LogicalDestination.</b> Reset: 0000h. Specifies one of up to sixteen x2APICs within the cluster specified by ClusterDestination. AccessType: X2APICEN ? Read-only,Error-on-write : Error-on-read,Error-on-write. <b>ValidValues:</b>																																		
	<table> <tr> <th>Bit</th><th>Description</th></tr> <tr><td>[0]</td><td>x2APIC 0</td></tr> <tr><td>[1]</td><td>x2APIC 1</td></tr> <tr><td>[2]</td><td>x2APIC 2</td></tr> <tr><td>[3]</td><td>x2APIC 3</td></tr> <tr><td>[4]</td><td>x2APIC 4</td></tr> <tr><td>[5]</td><td>x2APIC 5</td></tr> <tr><td>[6]</td><td>x2APIC 6</td></tr> <tr><td>[7]</td><td>x2APIC 7</td></tr> <tr><td>[8]</td><td>x2APIC 8</td></tr> <tr><td>[9]</td><td>x2APIC 9</td></tr> <tr><td>[10]</td><td>x2APIC 10</td></tr> <tr><td>[11]</td><td>x2APIC 11</td></tr> <tr><td>[12]</td><td>x2APIC 12</td></tr> <tr><td>[13]</td><td>x2APIC 13</td></tr> <tr><td>[14]</td><td>x2APIC 14</td></tr> <tr><td>[15]</td><td>x2APIC 15</td></tr> </table>	Bit	Description	[0]	x2APIC 0	[1]	x2APIC 1	[2]	x2APIC 2	[3]	x2APIC 3	[4]	x2APIC 4	[5]	x2APIC 5	[6]	x2APIC 6	[7]	x2APIC 7	[8]	x2APIC 8	[9]	x2APIC 9	[10]	x2APIC 10	[11]	x2APIC 11	[12]	x2APIC 12	[13]	x2APIC 13	[14]	x2APIC 14	[15]	x2APIC 15
Bit	Description																																		
[0]	x2APIC 0																																		
[1]	x2APIC 1																																		
[2]	x2APIC 2																																		
[3]	x2APIC 3																																		
[4]	x2APIC 4																																		
[5]	x2APIC 5																																		
[6]	x2APIC 6																																		
[7]	x2APIC 7																																		
[8]	x2APIC 8																																		
[9]	x2APIC 9																																		
[10]	x2APIC 10																																		
[11]	x2APIC 11																																		
[12]	x2APIC 12																																		
[13]	x2APIC 13																																		
[14]	x2APIC 14																																		
[15]	x2APIC 15																																		

**MSR0000\_080F [Spurious Interrupt Vector] (Core::X86::Msr::SVR)**

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_080F

Bits	Description
63:10	Reserved.
9	<b>FocusDisable.</b> Reset: 0. 1=Disable focus core checking during lowest-priority arbitrated interrupts. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.
8	<b>APICSWEn: APIC software enable.</b> Reset: 0. All LVT entry mask bits are set and cannot be cleared. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.
7:0	<b>Vector.</b> Reset: FFh. The vector that is sent to the core in the event of a spurious interrupt. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.

**MSR0000\_081[0...7] [In Service Register] (Core::X86::Msr::ISR)**

Reset: 0000\_0000\_0000\_0000h.

Interrupt In Service status bits [255:0] accessible through 8 ISR registers.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nISR0\_aliasMSR; MSR0000\_0810

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nISR1\_aliasMSR; MSR0000\_0811

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nISR2\_aliasMSR; MSR0000\_0812

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nISR3\_aliasMSR; MSR0000\_0813

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nISR4\_aliasMSR; MSR0000\_0814

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nISR5\_aliasMSR; MSR0000\_0815

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nISR6\_aliasMSR; MSR0000\_0816

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nISR7\_aliasMSR; MSR0000\_0817

Bits	Description
63:32	Reserved.
31:0	<b>InServiceBits.</b> Reset: 0000_0000h. These bits are set when the corresponding interrupt is being serviced by the core. AccessType: X2APICEN ? Read-only,Error-on-write,Volatile : Error-on-read,Error-on-write.

**MSR0000\_081[8...F] [Trigger Mode Register] (Core::X86::Msr::TMR)**

Reset: 0000\_0000\_0000\_0000h.

Trigger Mode status bits [255:0] accessible through 8 TMR registers.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nTMR0\_aliasMSR; MSR0000\_0818

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nTMR1\_aliasMSR; MSR0000\_0819

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nTMR2\_aliasMSR; MSR0000\_081A

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nTMR3\_aliasMSR; MSR0000\_081B

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nTMR4\_aliasMSR; MSR0000\_081C

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nTMR5\_aliasMSR; MSR0000\_081D

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nTMR6\_aliasMSR; MSR0000\_081E

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nTMR7\_aliasMSR; MSR0000\_081F

Bits	Description						
63:32	Reserved.						
31:0	<b>TriggerModeBits.</b> Reset: 0000_0000h. The corresponding trigger mode bit is updated when an interrupt is accepted. AccessType: X2APICEN ? Read-only,Error-on-write,Volatile : Error-on-read,Error-on-write. <b>ValidValues:</b>						
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0</td><td>Edge-triggered interrupt</td></tr> <tr> <td>1</td><td>Level-triggered interrupt</td></tr> </table>	Value	Description	0	Edge-triggered interrupt	1	Level-triggered interrupt
Value	Description						
0	Edge-triggered interrupt						
1	Level-triggered interrupt						

**MSR0000\_082[0...7] [Interrupt Request Register] (Core::X86::Msr::IRR)**

Reset: 0000\_0000\_0000\_0000h.

Interrupt Request status bits [255:0] accessible through 8 IRR registers.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nIRR0\_aliasMSR; MSR0000\_0820

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nIRR1\_aliasMSR; MSR0000\_0821

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nIRR2\_aliasMSR; MSR0000\_0822

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nIRR3\_aliasMSR; MSR0000\_0823

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nIRR4\_aliasMSR; MSR0000\_0824

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nIRR5\_aliasMSR; MSR0000\_0825

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nIRR6\_aliasMSR; MSR0000\_0826

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nIRR7\_aliasMSR; MSR0000\_0827

Bits	Description
63:32	Reserved.
31:0	<b>RequestBits.</b> Reset: 0000_0000h. The corresponding request bit is set when the an interrupt is accepted by the x2APIC. AccessType: X2APICEN ? Read-only,Error-on-write,Volatile : Error-on-read,Error-on-write.

**MSR0000\_0828 [Error Status Register] (Core::X86::Msr::ESR)**

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0828

Bits	Description
63:8	Reserved.
7	<b>IllegalRegAddr: illegal register address.</b> Reset: 0. This bit indicates that an access to a nonexistent register location within this APIC was attempted. Can only be set in xAPIC mode. AccessType: X2APICEN ? Read,Write-0-only,Error-on-write-1,Volatile : Error-on-read,Error-on-write.
6	<b>RcvdIllegalVector: received illegal vector.</b> Reset: 0. This bit indicates that this APIC has received a message with an illegal vector (00h to 0Fh for fixed and lowest priority interrupts). AccessType: X2APICEN ? Read,Write-0-only,Error-on-write-1,Volatile : Error-on-read,Error-on-write.
5	<b>SentIllegalVector.</b> Reset: 0. This bit indicates that this x2APIC attempted to send a message with an illegal vector (00h to 0Fh for fixed and lowest priority interrupts). AccessType: X2APICEN ? Read,Write-0-only,Error-on-write-1,Volatile : Error-on-read,Error-on-write.
4	Reserved.
3	<b>RcvAcceptError: receive accept error.</b> Reset: 0. This bit indicates that a message received by this APIC was not accepted by this or any other x2APIC. AccessType: X2APICEN ? Read,Write-0-only,Error-on-write-1,Volatile : Error-on-read,Error-on-write.
2	<b>SendAcceptError.</b> Reset: 0. This bit indicates that a message sent by this APIC was not accepted by any x2APIC. AccessType: X2APICEN ? Read,Write-0-only,Error-on-write-1,Volatile : Error-on-read,Error-on-write.
1:0	Reserved.

**MSR0000\_0830 [Interrupt Command] (Core::X86::Msr::InterruptCommand)**

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0830

Bits	Description																		
63:32	<b>DestinationField.</b> Reset: 0000_0000h. The destination encoding used when Core::X86::Msr::InterruptCommand[DestShrthnd] is 00b. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.																		
31:20	Reserved.																		
19:18	<b>DestShrthnd: destination shorthand.</b> Reset: 0h. Provides a quick way to specify a destination for a message. If all including self or all excluding self is used, then destination mode is ignored and physical is automatically used. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>No shorthand (Destination field).</td></tr> <tr> <td>1h</td><td>Self.</td></tr> <tr> <td>2h</td><td>All including self.</td></tr> <tr> <td>3h</td><td>All excluding self. (This sends a message with a destination encoding of all 1s, so if lowest priority is used the message could end up being reflected back to this APIC.)</td></tr> </table>	Value	Description	0h	No shorthand (Destination field).	1h	Self.	2h	All including self.	3h	All excluding self. (This sends a message with a destination encoding of all 1s, so if lowest priority is used the message could end up being reflected back to this APIC.)								
Value	Description																		
0h	No shorthand (Destination field).																		
1h	Self.																		
2h	All including self.																		
3h	All excluding self. (This sends a message with a destination encoding of all 1s, so if lowest priority is used the message could end up being reflected back to this APIC.)																		
17:16	Reserved.																		
15	<b>TM: trigger mode.</b> Reset: 0. 0=Edge triggered. 1=Level triggered. Indicates how this interrupt is triggered. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.																		
14	<b>Level.</b> Reset: 0. 0=Deasserted. 1=Asserted. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.																		
13:12	Reserved.																		
11	<b>DM: destination mode.</b> Reset: 0. 0=Physical. 1=Logical. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.																		
10:8	<b>MsgType.</b> Reset: 0h. The message types are encoded as follows: AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Fixed</td></tr> <tr> <td>1h</td><td>Lowest Priority.</td></tr> <tr> <td>2h</td><td>SMI</td></tr> <tr> <td>3h</td><td>Reserved.</td></tr> <tr> <td>4h</td><td>NMI</td></tr> <tr> <td>5h</td><td>INIT</td></tr> <tr> <td>6h</td><td>Startup</td></tr> <tr> <td>7h</td><td>External interrupt.</td></tr> </table>	Value	Description	0h	Fixed	1h	Lowest Priority.	2h	SMI	3h	Reserved.	4h	NMI	5h	INIT	6h	Startup	7h	External interrupt.
Value	Description																		
0h	Fixed																		
1h	Lowest Priority.																		
2h	SMI																		
3h	Reserved.																		
4h	NMI																		
5h	INIT																		
6h	Startup																		
7h	External interrupt.																		
7:0	<b>Vector.</b> Reset: 00h. The vector that is sent for this interrupt source. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.																		



**MSR0000\_0832 [LVT Timer] (Core::X86::Msr::TimerLvtEntry)**

Reset: 0000\_0000\_0001\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0832

Bits	Description
63:18	Reserved.
17	<b>Mode.</b> Reset: 0. 0=One-shot. 1=Periodic. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.
16	<b>Mask.</b> Reset: 1. 0=Not masked. 1=Masked. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.
15:13	Reserved.
12	<b>DS: interrupt delivery status.</b> Reset: 0. 0=Idle. 1=Send pending. (Indicates that the interrupt has not yet been accepted by the core.) AccessType: X2APICEN ? Read-only,Volatile : Error-on-read,Error-on-write.
11:8	Reserved.
7:0	<b>Vector.</b> Reset: 00h. Interrupt vector number. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.

**MSR0000\_0833 [LVT Thermal Sensor] (Core::X86::Msr::ThermalLvtEntry)**

Reset: 0000\_0000\_0001\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0833

Bits	Description
63:17	Reserved.
16	<b>Mask.</b> Reset: 1. 0=Not masked. 1=Masked. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.
15:13	Reserved.
12	<b>DS: interrupt delivery status.</b> Reset: 0. 0=Idle. 1=Send pending. (Indicates that the interrupt has not yet been accepted by the core.) AccessType: X2APICEN ? Read-only,Volatile : Error-on-read,Error-on-write.
11	Reserved.
10:8	<b>MsgType: message type.</b> Reset: 0h. See 2.1.11.2.1.14 [Generalized Local Vector Table]. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.
7:0	<b>Vector.</b> Reset: 00h. Interrupt vector number. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.

**MSR0000\_0834 [LVT Performance Monitor] (Core::X86::Msr::PerformanceCounterLvtEntry)**

Reset: 0000\_0000\_0001\_0000h.

Interrupts for this local vector table are caused by overflows of:

- Core::X86::Msr::PERF\_LEGACY\_CTL0..3(Performance Event Select [3:0]).
- Core::X86::Msr::PERF\_CTL0..5(Performance Event Select [5:0]).

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0834

Bits	Description
63:17	Reserved.
16	<b>Mask.</b> Reset: 1. 0=Not masked. 1=Masked. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.
15:13	Reserved.
12	<b>DS: interrupt delivery status.</b> Reset: 0. 0=Idle. 1=Send pending. Indicates that the interrupt has not yet been accepted by the core. AccessType: X2APICEN ? Read-only,Volatile : Error-on-read,Error-on-write.
11	Reserved.
10:8	<b>MsgType: message type.</b> Reset: 0h. See2.1.11.2.1.14 [Generalized Local Vector Table]. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.
7:0	<b>Vector.</b> Reset: 00h. Interrupt vector number. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.

**MSR0000\_083[5...6] [LVT LINT[1:0]] (Core::X86::Msr::LVTINT)**

Reset: 0000\_0000\_0001\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nLVTINT0\_aliasMSR; MSR0000\_0835

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_nLVTINT1\_aliasMSR; MSR0000\_0836

Bits	Description
63:17	Reserved.
16	<b>Mask.</b> Reset: 1. 0=Not masked. 1=Masked. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.
15	<b>TM: trigger mode.</b> Reset: 0. 0=Edge. 1=Level. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.
14	<b>RmtIRR.</b> Reset: 0. If trigger mode is level, remote Core::X86::Msr::IRR is set when the interrupt has begun service. Remote Core::X86::Msr::IRR is cleared when the end of interrupt has occurred. AccessType: X2APICEN ? Read-only,Volatile : Error-on-read,Error-on-write.
13	Reserved.
12	<b>DS: interrupt delivery status.</b> Reset: 0. 0=Idle. 1=Send pending. (Indicates that the interrupt has not yet been accepted by the core.) AccessType: X2APICEN ? Read-only,Volatile : Error-on-read,Error-on-write.
11	Reserved.
10:8	<b>MsgType: message type.</b> Reset: 0h. See2.1.11.2.1.14 [Generalized Local Vector Table]. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.
7:0	<b>Vector.</b> Reset: 00h. Interrupt vector number. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.

**MSR0000\_0837 [LVT Error] (Core::X86::Msr::ErrorLvtEntry)**

Reset: 0000\_0000\_0001\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0837

Bits	Description
63:17	Reserved.
16	<b>Mask.</b> Reset: 1. 0=Not masked. 1=Masked. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.
15:13	Reserved.
12	<b>DS: interrupt delivery status.</b> Reset: 0. 0=Idle. 1=Send pending. (Indicates that the interrupt has not yet been accepted by the core.) AccessType: X2APICEN ? Read-only, Volatile : Error-on-read,Error-on-write.
11	Reserved.
10:8	<b>MsgType: message type.</b> Reset: 0h. See 2.1.11.2.1.14 [Generalized Local Vector Table]. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.
7:0	<b>Vector.</b> Reset: 00h. Interrupt vector number. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.

**MSR0000\_0838 [Timer Initial Count] (Core::X86::Msr::TimerInitialCount)**

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0838

Bits	Description
63:32	Reserved.
31:0	<b>Count.</b> Reset: 0000_0000h. The value copied into the current count register when the timer is loaded or reloaded. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.

**MSR0000\_0839 [Timer Current Count] (Core::X86::Msr::TimerCurrentCount)**

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0839

Bits	Description
63:32	Reserved.
31:0	<b>Count.</b> Reset: 0000_0000h. The current value of the counter. AccessType: X2APICEN ? Read,Error-on-write,Volatile : Error-on-read,Error-on-write.

**MSR0000\_083E [Timer Divide Configuration] (Core::X86::Msr::TimerDivideConfiguration)**

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_083E

Bits	Description
63:4	Reserved.
3:0	<b>Div[3:0]</b> . Reset: 0h. Div[2] is unused. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.
<b>ValidValues:</b>	
Value	Description
0h	Divide by 2.
1h	Divide by 4.
2h	Divide by 8.
3h	Divide by 16.
7h-4h	Reserved.
8h	Divide by 32.
9h	Divide by 64.
Ah	Divide by 128.
Bh	Divide by 1.
Fh-Ch	Reserved.

**MSR0000\_083F [Self IPI] (Core::X86::Msr::SelfIPI)**

Reset: 0000\_0000\_0000\_0000h.

The self IPI register provides a performance optimized path for sending self IPI's. A self IPI is semantically identical to an inter-processor interrupt sent via the ICR, with a Destination Shorthand of Self, Trigger Mode equal to Edge, and a Delivery Mode equal to Fixed.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_083F

Bits	Description
63:8	Reserved.
7:0	<b>Vector</b> . Reset: 00h. Interrupt vector number. AccessType: X2APICEN ? Write-only,Error-on-read : Error-on-read,Error-on-write.

**MSR0000\_0840 [Extended APIC Feature] (Core::X86::Msr::ExtendedApicFeature)**

Reset: 0000\_0000\_0004\_0007h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0840

Bits	Description
63:24	Reserved.
23:16	<b>ExtLvtCount: extended local vector table count</b> . Reset: 04h. This specifies the number of extended LVT registers (Core::X86::Msr::ExtendedInterruptLvtEntries) in the local APIC. AccessType: X2APICEN ? Read-only,Error-on-write : Error-on-read,Error-on-write.
15:3	Reserved.
2	<b>ExtApicIdCap: extended APIC ID capable</b> . Reset: 1. 1=The processor is capable of supporting an 8-bit APIC ID, as controlled by Core::X86::Msr::ExtendedApicControl[ExtApicIdEn]. AccessType: X2APICEN ? Read-only,Error-on-write : Error-on-read,Error-on-write.
1	<b>SeoiCap: specific end of interrupt capable</b> . Reset: 1. 1=The Core::X86::Msr::SpecificEndOfInterrupt is present. AccessType: X2APICEN ? Read-only,Error-on-write : Error-on-read,Error-on-write.
0	<b>IerCap: interrupt enable register capable</b> . Reset: 1. This bit indicates that the Core::X86::Msr::InterruptEnable0 - 7 are present. See 2.1.11.2.1.8 [Interrupt Masking]. AccessType: X2APICEN ? Read-only,Error-on-write : Error-on-read,Error-on-write.

**MSR0000\_0841 [Extended APIC Control] (Core::X86::Msr::ExtendedApicControl)**

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0841

Bits	Description
63:3	Reserved.
2	<b>ExtApicIdEn: extended APIC ID enable.</b> Reset: 0. 1=Enable 8-bit APIC ID; Core::X86::Msr::APIC_ID[ApicId[31:0]] supports an 8-bit value; an interrupt broadcast in physical destination mode requires that the (IntDest[7:0] = 1111_1111b) (instead of XXXX_1111b); a match in physical destination mode occurs when (IntDest[7:0] == ApicId[7:0]) instead of (IntDest[3:0] == ApicId[3:0]).
	AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.
1	<b>SeoiEn.</b> Reset: 0. 1=Enable SEOI generation when a write to Core::X86::Msr::SpecificEndOfInterrupt is received.
	AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.
0	<b>IerEn.</b> Reset: 0. 1=Enable writes to the interrupt enable registers.
	AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.

**MSR0000\_0842 [Specific End Of Interrupt] (Core::X86::Msr::SpecificEndOfInterrupt)**

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0842

Bits	Description
63:8	Reserved.
7:0	<b>EoiVec: end of interrupt vector.</b> Reset: 00h. A write to this field causes an end of interrupt cycle to be performed for the vector specified in this field. The behavior is undefined if no interrupt is pending for the specified interrupt vector.
	AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.

**MSR0000\_0848 [Interrupt Enable 0] (Core::X86::Msr::InterruptEnable0)**

Reset: 0000\_0000\_FFFF\_FFFFh.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n0\_aliasMSR; MSR0000\_0848

Bits	Description
63:32	Reserved.
31:16	<b>InterruptEnableBits.</b> Reset: FFFFh. The interrupt enable bits can be used to enable each of the 256 interrupts.
	AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.
15:0	Reserved.

**MSR0000\_084[9...F] [Interrupt Enable 7..1] (Core::X86::Msr::InterruptEnable71)**

Reset: 0000\_0000\_FFFF\_FFFFh.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n1\_aliasMSR; MSR0000\_0849

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n2\_aliasMSR; MSR0000\_084A

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n3\_aliasMSR; MSR0000\_084B

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n4\_aliasMSR; MSR0000\_084C

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n5\_aliasMSR; MSR0000\_084D

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n6\_aliasMSR; MSR0000\_084E

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n7\_aliasMSR; MSR0000\_084F

Bits	Description
63:32	Reserved.
31:0	<b>InterruptEnableBits.</b> Reset: FFFF_FFFFh. The interrupt enable bits can be used to enable each of the 256 interrupts.
	AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.

**MSR0000\_085[0...3] [Extended Interrupt Local Vector Table] (Core::X86::Msr::ExtendedInterruptLvtEntries)**

Reset: 0000\_0000\_0001\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n0\_aliasMSR; MSR0000\_0850

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n1\_aliasMSR; MSR0000\_0851

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n2\_aliasMSR; MSR0000\_0852

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n3\_aliasMSR; MSR0000\_0853

Bits	Description
63:17	Reserved.
16	<b>Mask.</b> Reset: 1. 0=Not masked. 1=Masked. Interrupt Mask. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.
15:13	Reserved.
12	<b>DS: interrupt delivery status.</b> Reset: 0. 0=Idle. 1=Send pending. Indicates that the interrupt has not yet been accepted by the core. AccessType: X2APICEN ? Read-write, Volatile : Error-on-read,Error-on-write.
11	Reserved.
10:8	<b>MsgType: message type.</b> Reset: 0h. See 2.1.11.2.1.14 [Generalized Local Vector Table]. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.
7:0	<b>Vector.</b> Reset: 00h. Interrupt vector number. AccessType: X2APICEN ? Read-write : Error-on-read,Error-on-write.

**MSR0000\_0C8D [Monitoring Event Select] (Core::X86::Msr::QM\_EVTSEL)**

\_lthree[1:0]; MSR0000\_0C8D

Bits	Description
63:44	Reserved.
43:32	<b>RMID.</b> Read-write. Reset: 000h. Resource Monitoring Identifier.
31	<b>ExtendedEvtID.</b> Read-write. Reset: 0. When set, the EventId fields refers to QoS Extended Feature Identifiers.
30:8	Reserved.
7:0	<b>EventId.</b> Read-write. Reset: 00h. Monitored Event ID.

**MSR0000\_0C8E [QOS L3 Counter] (Core::X86::Msr::QM\_CTR)**

Read,Error-on-write. Reset: 8000\_0000\_0000\_0000h.

\_lthree[1:0]; MSR0000\_0C8E

Bits	Description
63	<b>Error.</b> Read,Error-on-write. Reset: 1. Unsupported RMID or event type was written to Core::X86::Msr::QM_EVTSEL.
62	<b>Unavailable.</b> Read,Error-on-write. Reset: 0. Data for this RMID is not available or not monitored for this resource or RMID.
61:0	<b>RmData.</b> Read,Error-on-write. Reset: 0000_0000_0000_0000h. Resource Monitored Data.

**MSR0000\_0C8F (Core::X86::Msr::PQR\_ASSOC)**

Reset: 0000\_0000\_0000\_0000h.

QOS L2 RMID. The behavior of this register is defined in the AMD64 Technology Platform Quality of Service Extensions specification.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSR0000\_0C8F

Bits	Description
63:36	Reserved.
35:32	<b>Clos.</b> Read-write. Reset: 0h. Class of Service.
31:12	Reserved.
11:0	<b>Rmid.</b> Read-write. Reset: 000h. Resource Monitor Identifier.

**MSR0000\_0DA0 [Extended Supervisor State] (Core::X86::Msr::XSS)**

_lthree[1:0]_core[7:0]_thread[1:0]; MSR0000_0DA0	
Bits	Description
63:13	Reserved.
12	<b>CET_S</b> . Read-write. Reset: 0. System Control-flow Enforcement Technology.
11	<b>CET_U</b> . Read-write. Reset: 0. User Control-flow Enforcement Technology.
10:0	Reserved.

**2.1.13.2 MSRs – MSRC000\_xxxx****MSRC000\_0080 [Extended Feature Enable] (Core::X86::Msr::EFER)**

SKINIT Execution: 0000_0000_0000_0000h.	
_lthree[1:0]_core[7:0]_thread[1:0]; MSRC000_0080	
Bits	Description
63:22	Reserved.
21	<b>AutomaticIBRSEn: Automatic IBRS Enable</b> . Read-write. Reset: 0. 0=IBRS protection is not enabled unless (SPEC_CTRL[IBRS] == 1). 1=IBRS protection is enabled for any process running at (CPL == 0) or ((ASID == 0) && SEV-SNP).
20	<b>UAIE: Upper Address Ignore Enable</b> . Read-write. Reset: 0. Upper Address Ignore suppresses canonical faults for most data access virtual addresses, which allows software to use the upper bits of a virtual address as tags.
19	Reserved.
18	<b>IntWbinvdEn</b> . Read-write. Reset: 0. Interruptible wbinvd, wbinvd enable.
17:16	Reserved.
15	<b>TCE: translation cache extension enable</b> . Read-write. Reset: 0. 1=Translation cache extension is enabled. PDC entries related to the linear address of the INVLPG instruction are invalidated. If this bit is 0 all PDC entries are invalidated by the INVLPG instruction.
14	<b>FFXSE: fast FXSAVE/FRSTOR enable</b> . Read-write. Reset: 0. 1=Enables the fast FXSAVE/FRSTOR mechanism. A 64-bit operating system may enable the fast FXSAVE/FRSTOR mechanism if (Core::X86::CpuId::FeatureExtIdEdx[FFXSR] == 1). This bit is set once by the operating system and its value is not changed afterwards.
13	<b>LMSLE: long mode segment limit enable</b> . Read-only, Error-on-write-1. Reset: Fixed, 0. 1=Enables the long mode segment limit check mechanism.
12	<b>SVME: secure virtual machine (SVM) enable</b> . Reset: Fixed, 0. 1=SVM features are enabled. AccessType: Core::X86::Msr::VM_CR[SvmeDisable] ? Read-only, Error-on-write-1 : Read-write.
11	<b>NXE: no-execute page enable</b> . Read-write. Reset: 0. 1=The no-execute page protection feature is enabled.
10	<b>LMA: long mode active</b> . Read-only. Reset: 0. 1=Indicates that long mode is active. When writing the EFER register the value of this bit must be preserved. Software must read the EFER register to determine the value of LMA, change any other bits as required and then write the EFER register. An attempt to write a value that differs from the state determined by hardware results in a #GP fault.
9	Reserved.
8	<b>LME: long mode enable</b> . Read-write. Reset: 0. 1=Long mode is enabled.
7:1	Reserved.
0	<b>SYSCALL: system call extension enable</b> . Read-write. Reset: 0. 1=SYSCALL and SYSRET instructions are enabled. This adds the SYSCALL and SYSRET instructions which can be used in flat addressed operating systems as low latency system calls and returns.

**MSRC000\_0081 [SYSCALL Target Address] (Core::X86::Msr::STAR)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

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

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_0081

Bits	Description
63:48	<b>SysRetSel.</b> Read-write. Reset: 0000h. SYSRET CS and SS.
47:32	<b>SysCallSel.</b> Read-write. Reset: 0000h. SYSCALL CS and SS.
31:0	<b>Target.</b> Read-write. Reset: 0000_0000h. SYSCALL target address.

**MSRC000\_0082 [Long Mode SYSCALL Target Address] (Core::X86::Msr::STAR64)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_0082

Bits	Description
63:0	<b>LSTAR: long mode target address.</b> Read-write. Reset: 0000_0000_0000_0000h. Target address for 64-bit mode calling programs. The address stored in this register must be in canonical form (if not canonical, a #GP fault occurs).

**MSRC000\_0083 [Compatibility Mode SYSCALL Target Address] (Core::X86::Msr::STARCOMPAT)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_0083

Bits	Description
63:0	<b>CSTAR: compatibility mode target address.</b> Read-write. Reset: 0000_0000_0000_0000h. Target address for compatibility mode. The address stored in this register must be in canonical form (if not canonical, a #GP fault occurs).

**MSRC000\_0084 [SYSCALL Flag Mask] (Core::X86::Msr::SYSCALL\_FLAG\_MASK)**

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_0084

Bits	Description
63:32	Reserved.
31:0	<b>Mask: SYSCALL flag mask.</b> Read-write. Reset: 0000_0000h. This register holds the EFLAGS mask used by the SYSCALL instruction. 1=Clear the corresponding EFLAGS bit when executing the SYSCALL instruction.

**MSRC000\_00E7 [Read-Only Max Performance Frequency Clock Count] (Core::X86::Msr::MPerfReadOnly)**

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_00E7

Bits	Description
63:0	<b>MPerfReadOnly: read-only maximum core clocks counter.</b> Reset: 0000_0000_0000_0000h. Incremented by hardware at the P0 frequency while the core is in C0. In combination with Core::X86::Msr::APerfReadOnly, this is used to determine the effective frequency of the core. A read of this MSR in guest mode is affected by Core::X86::Msr::TscRateMsr. This field uses software P-state numbering. See Core::X86::Msr::HWCR[EffFreqCntMwait], 2.1.4 [Effective Frequency]. This register is not affected by writes to Core::X86::Msr::MPERF. AccessType: Core::X86::Msr::HWCR[EffFreqReadOnlyLock] ? Read,Error-on-write,Volatile : Read-write,Volatile.



**MSRC000\_00E8 [Read-Only Actual Performance Frequency Clock Count] (Core::X86::Msr::APerfReadOnly)**

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_00E8

Bits	Description
63:0	<b>APerfReadOnly: read-only actual core clocks counter.</b> Reset: 0000_0000_0000_0000h. This register increments in proportion to the actual number of core clocks cycles while the core is in C0. See Core::X86::Msr::MPerfReadOnly. This register is not affected by writes to Core::X86::Msr::APERF. AccessType: Core::X86::Msr::HWCR[EffFreqReadOnlyLock] ? Read,Error-on-write,Volatile : Read-write,Volatile.

**MSRC000\_00E9 [Instructions Retired Performance Count] (Core::X86::Msr::IRPerfCount)**

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_00E9

Bits	Description
63:48	Reserved.
47:0	<b>IRPerfCount: instructions retired counter.</b> Reset: 0000_0000_0000_0000h. Dedicated Instructions Retired register increments on once for every instruction retired. See Core::X86::Msr::HWCR[IRPerfEn]. AccessType: Core::X86::Msr::HWCR[EffFreqReadOnlyLock] ? Read,Error-on-write,Volatile : Read-write,Volatile.

**MSRC000\_0100 [FS Base] (Core::X86::Msr::FS\_BASE)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_0100

Bits	Description
63:0	<b>FSBase: expanded FS segment base.</b> Read-write. Reset: 0000_0000_0000_0000h. This register provides access to the expanded 64-bit FS segment base. The address stored in this register must be in canonical form (if not canonical, a #GP fault fill occurs).

**MSRC000\_0101 [GS Base] (Core::X86::Msr::GS\_BASE)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_0101

Bits	Description
63:0	<b>GSBase: expanded GS segment base.</b> Read-write. Reset: 0000_0000_0000_0000h. This register provides access to the expanded 64-bit GS segment base. The address stored in this register must be in canonical form (if not canonical, a #GP fault fill occurs).

**MSRC000\_0102 [Kernel GS Base] (Core::X86::Msr::KernelGSbase)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_0102

Bits	Description
63:0	<b>KernelGSBase: kernel data structure pointer.</b> Read-write. Reset: 0000_0000_0000_0000h. This register holds the kernel data structure pointer which can be swapped with the GS_BASE register using the SwapGS instruction. The address stored in this register must be in canonical form (if not canonical, a #GP fault occurs).

**MSRC000\_0103 [Auxiliary Time Stamp Counter] (Core::X86::Msr::TSC\_AUX)**

Read-write,Volatile. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_0103

Bits	Description
63:32	Reserved.
31:0	<b>TscAux: auxiliary time stamp counter data.</b> Read-write,Volatile. Reset: 0000_0000h. It is expected that this is initialized by privileged software to a meaningful value, such as a processor ID. This value is returned in the RDTSCP instruction.

**MSRC000\_0104 [Time Stamp Counter Ratio] (Core::X86::Msrr::TscRateMsrr)**

Core::X86::Msrr::TscRateMsrr allows the hypervisor to control the guest's view of the Time Stamp Counter. It provides a multiplier that scales the value returned when Core::X86::Msrr::TSC[TSC], Core::X86::Msrr::MPERF[MPERF], and Core::X86::Msrr::MPerfReadOnly[MPerfReadOnly] are read by a guest running under virtualization. This allows the hypervisor to provide a consistent TSC, MPERF, and MPerfReadOnly rate for a guest process when moving that process between cores that have a differing P0 rate. The TSC Ratio MSR does not affect the value read from the TSC, MPERF, and MPerfReadOnly MSRs when read when in host mode or when virtualization is not being used or when accessed by code executed in system management mode (SMM) unless the SMM code is executed within a guest container. The TSC Ratio value does not affect the rate of the underlying TSC, MPERF, and MPerfReadOnly counters, or the value that gets written to the TSC, MPERF, and MPerfReadOnly MSRs counters on a Write by either the host or the guest. The TSC Ratio MSR contains a fixed-point number in 8.32 format, which is 8 bits of integer and 32 bits of fraction. This number is the ratio of the desired P0 frequency to the P0 frequency of the core. The reset value of the TSC Ratio MSR is 1.0, which results when a guest frequency matches the core P0 frequency.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_0104

Bits	Description
63:40	Reserved.
39:32	<b>TscRateMsrrInt: time stamp counter rate integer.</b> Read-write. Reset: 01h. Specifies the integer part of the MSR TSC ratio value.
31:0	<b>TscRateMsrrFrac: time stamp counter rate fraction.</b> Read-write. Reset: 0000_0000h. Specifies the fractional part of the MSR TSC ratio value.

**MSRC000\_0108 [Prefetch Control] (Core::X86::Msrr::PrefetchControl)**

Reset: 0000\_0000\_0000\_03C0h.

\_lthree[1:0]\_core[7:0]; MSRC000\_0108

Bits	Description														
63:10	Reserved.														
9:7	<b>PrefetchAggressivenessProfile.</b> Read-write. Reset: 7h. When MasterEnable is set, selects a prefetch aggressiveness profile. <b>ValidValues:</b>														
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Level 0, least aggressive prefetch profile.</td></tr> <tr> <td>1h</td><td>Level 1</td></tr> <tr> <td>2h</td><td>Level 2</td></tr> <tr> <td>3h</td><td>Level 3, most aggressive prefetch profile.</td></tr> <tr> <td>6h-4h</td><td>Reserved.</td></tr> <tr> <td>7h</td><td>Default used by hardware. Not software accessible.</td></tr> </table>	Value	Description	0h	Level 0, least aggressive prefetch profile.	1h	Level 1	2h	Level 2	3h	Level 3, most aggressive prefetch profile.	6h-4h	Reserved.	7h	Default used by hardware. Not software accessible.
Value	Description														
0h	Level 0, least aggressive prefetch profile.														
1h	Level 1														
2h	Level 2														
3h	Level 3, most aggressive prefetch profile.														
6h-4h	Reserved.														
7h	Default used by hardware. Not software accessible.														
6	<b>MasterEnable.</b> Read-write. Reset: 1. Enable prefetch aggressiveness profiles.														
5	<b>UpDown.</b> Read-write. Reset: 0. Disable prefetcher that uses memory access history to determine whether to fetch the next or previous line into L2 cache for all memory accesses.														
4	Reserved.														
3	<b>L2Stream.</b> Read-write. Reset: 0. Disable prefetcher that uses history of memory access patterns to fetch additional sequential lines into L2 cache.														
2	<b>L1Region.</b> Read-write. Reset: 0. Disable prefetcher that uses memory access history to fetch additional lines into L1 cache when the data access for a given instruction tends to be followed by a consistent pattern of other accesses within a localized region.														
1	<b>L1Stride.</b> Read-write. Reset: 0. Disable stride prefetcher that uses memory access history of individual instructions to fetch additional lines into L1 cache when each access is a constant distance from the previous.														
0	<b>L1Stream.</b> Read-write. Reset: 0. Disable stream prefetcher that uses history of memory access patterns to fetch additional sequential lines into L1 cache.														

**MSRC000\_010E [Last Branch Stack Select] (Core::X86::Msr::LastBranchStackSelect)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

This MSR allows Last Branch Stack recording to be suppressed based on branch type and privilege level.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_010E

Bits	Description
63:9	Reserved.
8	<b>FarBranch.</b> Read-write. Reset: 0. When set far branches are not recorded.
7	<b>JmpNearRel.</b> Read-write. Reset: 0. When set, near relative jumps, excluding near relative calls, are not recorded.
6	<b>JmpNearInd.</b> Read-write. Reset: 0. When set, near indirect jumps, excluding near indirect calls and near returns, are not recorded.
5	<b>RetNear.</b> Read-write. Reset: 0. When set, near returns are not recorded.
4	<b>CallNearInd.</b> Read-write. Reset: 0. When set, near indirect calls are not recorded.
3	<b>CallNearRel.</b> Read-write. Reset: 0. When set, near relative calls are not recorded.
2	<b>Jcc.</b> Read-write. Reset: 0. When set conditional branches are not recorded.
1	<b>CplGe0.</b> Read-write. Reset: 0. When set, no branches ending in CPL > 0 are recorded.
0	<b>CplEq0.</b> Read-write. Reset: 0. When set, no branches ending in CPL = 0 are recorded.

**MSRC000\_010F [Debug Extension Control] (Core::X86::Msr::DebugExtnCtl)**

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_010F

Bits	Description
63:7	Reserved.
6	<b>LBRV2EN.</b> Read-write. Reset: 0. When enabled the last 16 branch targets and from addresses are recorded in Core::X86::Msr::LastBranchStackToIp and Core::X86::Msr::LastBranchStackFromIp. Core::X86::Msr::DebugExtnCtl[LBRV2EN] is cleared upon #DB entry.
5:0	Reserved.

**MSRC000\_0300 [Performance Counter Global Status] (Core::X86::Msr::PerfCntrGlobalStatus)**

Read-only. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_0300

Bits	Description
63:60	Reserved.
59	<b>PmcFreeze: Performance Counter Freeze.</b> Read-only. Reset: 0. When set, the performance monitor counters have been frozen due an interrupt signaled for the overflow of at least one counter. This feature is enabled via Core::X86::Msr::DBG_CTL_MSR[FPMCI].
58	<b>LbrFreeze: LBR Stack Freeze.</b> Read-only. Reset: 0. When set, LBR stack has been frozen due to an interrupt signaled for the overflow of at least one counter. This feature is enabled via Core::X86::Msr::DBG_CTL_MSR[FLBRI]. LBR Freeze does not affect the legacy LBR registers.
57:6	Reserved.
5:0	<b>PerfCntrOvfl: Performance Counter Overflow Bits.</b> Read-only. Reset: 00h. For each available core Performance Counter there is one overflow bit starting at bit position 0 for counter 0 (PerfCntr0). The bit is set when the corresponding counter overflows and remains set until cleared by software via Core::X86::Msr::PerfCntrGlobalStatusClr. The bits can also be set directly by software via Core::X86::Msr::PerfCntrGlobalStatusSet. Note that the overflow bit is set even when the Performance Counter was not configured to signal an interrupt.

**MSRC000\_0301 [Performance Counter Global Control] (Core::X86::Msr::PerfCntGlobalCtl)**

Read-write. Reset: 0000\_0000\_0000\_003Fh.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_0301

Bits	Description
63:6	Reserved.
5:0	<b>PerfCntEn: Global Performance Counter Enable.</b> Read-write. Reset: 3Fh. For each available performance counter there is one enable bit in this field. Bit position 0 corresponds to counter 0 (PerfCnt0), bit position 1 corresponds to counter 1 (PerfCnt1) and so forth. A Performance counter is enabled to count when both its Core::X86::Msr::PerfCntGlobalCtl[PerfCntEn] bit and its Core::X86::Msr::PERF_CTL0..5[En] bit are set.

**MSRC000\_0302 [Performance Counter Global Status Clear] (Core::X86::Msr::PerfCntGlobalStatusClr)**

Write-only. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_0302

Bits	Description
63:60	Reserved.
59	<b>PmcFreezeClr: PMC Freeze Clear.</b> Write-only. Reset: 0. When written as 1, Core::X86::Msr::PerfCntGlobalStatus[PmcFreeze] is cleared and the Performance Monitor Counter Freeze is cleared and counters that are enabled to count continue to count. Software should use this feature to restart counters after it serviced overflow conditions in a Performance Monitor interrupt handler.
58	<b>LbrFreezeClr: LBR Freeze Clear.</b> Write-only. Reset: 0. When written as 1, Core::X86::Msr::PerfCntGlobalStatus[LbrFreeze] is cleared, the LBR stack freeze is lifted and continues to record branches as configured. Software should use this feature to restart LBR after it serviced overflow conditions in a Performance Monitor interrupt handler.
57:6	Reserved.
5:0	<b>PerfCntOvflClr: Performance Counter Overflow Bits Clear.</b> Write-only. Reset: 00h. <b>Description:</b> These bits allow software to clear PerfCntOvfl bits in Core::X86::Msr::PerfCntGlobalStatus. To clear a bit software needs to write a 1 to the corresponding bit. Software should clear the Performance Counter Overflow bits when: <ul style="list-style-type: none"> <li>- Handling a performance counter overflow interrupt</li> <li>- Disabling a performance counter</li> <li>- Resetting a performance counter</li> </ul>

**MSRC000\_0303 [Performance Counter Global Status Set] (Core::X86::Msr::PerfCntGlobalStatusSet)**

Write-only. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_0303

Bits	Description
63:60	Reserved.
59	<b>PmcFreezeSet: PMC Freeze Set.</b> Write-only. Reset: 0. When written as 1, Core::X86::Msr::PerfCntGlobalStatus[PmcFreeze] is set and the Performance Monitor Counters are frozen. In normal operation it is not expected that software would need to set these bits. It is the hardware that sets this bit when Core::X86::Msr::DBG_CTL_MSR[FPMCI] is set and a PMC overflow signals an interrupt.
58	<b>LbrFreezeSet: LBR Freeze Set.</b> Write-only. Reset: 0. When written as 1, Core::X86::Msr::PerfCntGlobalStatus[LbrFreeze] is set, the LBR stack is frozen and no longer records branches. In normal operation it is not expected that software would need to set these bits. It is the hardware that sets this bit when Core::X86::Msr::DBG_CTL_MSR[FLBRI] is set and a PMC overflow signals an interrupt.
57:6	Reserved.
5:0	<b>PerfCntOvflSet: Performance Counter Overflow Bits Set.</b> Write-only. Reset: 00h. These bits allow software to set PerfCntOvfl bits PerfCntGlobalStatus. Setting an overflow bit in PerfCntGlobalStatus does not result in the generation of an interrupt, freeze of performance counters, freeze of LBR or other actions that may be taken when a performance counter overflows.

**MSRC000\_0410 [MCA Interrupt Configuration] (Core::X86::Msr::McaIntrCfg)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

MSRC000\_0410

Bits	Description
63:16	Reserved.
15:12	<b>ThresholdLvtOffset.</b> Read-write. Reset: 0h. For error thresholding interrupts, specifies the address of the LVT entry in the APIC registers as follows: LVT address = (LvtOffset shifted left 4 bits) + 500h (see Core::X86::Apic::ExtendedInterruptLvtEntries).
11:8	Reserved.
7:4	<b>DeferredLvtOffset.</b> Read-write. Reset: 0h. For deferred error interrupts, specifies the address of the LVT entry in the APIC registers as follows: LVT address = (LvtOffset shifted left 4 bits) + 500h (see APIC[530:500]).
3:0	Reserved.

**MSRC000\_0500 [Workload Classification Configuration Register] (Core::X86::Msr::WctCfg)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_0500

Bits	Description
63:1	Reserved.
0	<b>Enable.</b> Read-write. Reset: 0. When set, enables Workload Classification. When software clears the Enable bit, hardware clears the profiling history accumulated on that logical core.

**MSRC000\_0501 [Workload Classification Status Register] (Core::X86::Msr::WctStatus)**

Read-only. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_0501

Bits	Description
63	<b>Valid.</b> Read-only. Reset: 0. When set, indicates that ClassID is valid.
62:3	Reserved.
2:0	<b>ClassID.</b> Read-only. Reset: 0h. Workload Class ID. This value represents an index into a memory mapped table which contains scheduling hints. The Class ID is based on workload history since the last history reset.

**MSRC000\_0502 [History Reset] (Core::X86::Msr::HistoryReset)**

Write-only. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC000\_0502

Bits	Description
63:1	Reserved.
0	<b>WCHR.</b> Write-only. Reset: 0. When written as 1, Workload Classification History is reset.

**2.1.13.3 MSRs - MSRC001\_0xxx**

**MSRC001\_0000 [Performance Event Select 0] (Core::X86::Msr::PERF\_LEGACY\_CTL0)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

The legacy alias of Core::X86::Msr::PERF\_CTL0. See Core::X86::Msr::PERF\_CTL0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0000

Bits	Description										
63:42	Reserved.										
41:40	<b>HostGuestOnly: count only host/guest events.</b> Read-write. Reset: 0h.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Count all events, irrespective of guest/host.</td></tr> <tr> <td>1h</td><td>Count guest events if [SVME] == 1.</td></tr> <tr> <td>2h</td><td>Count host events if [SVME] == 1.</td></tr> <tr> <td>3h</td><td>Count all guest and host events if [SVME] == 1.</td></tr> </table>	Value	Description	0h	Count all events, irrespective of guest/host.	1h	Count guest events if [SVME] == 1.	2h	Count host events if [SVME] == 1.	3h	Count all guest and host events if [SVME] == 1.
Value	Description										
0h	Count all events, irrespective of guest/host.										
1h	Count guest events if [SVME] == 1.										
2h	Count host events if [SVME] == 1.										
3h	Count all guest and host events if [SVME] == 1.										
39:36	Reserved.										
35:32	<b>EventSelect[11:8].</b> Read-write. Reset: 0h. Performance event select[11:8].										
31:24	<b>CntMask: counter mask.</b> Read-write. Reset: 00h. Controls the number of events counted per clock cycle.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00h</td><td>The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.</td></tr> <tr> <td>7Fh-01h</td><td>When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.</td></tr> <tr> <td>FFh-80h</td><td>Reserved.</td></tr> </table>	Value	Description	00h	The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.	7Fh-01h	When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.	FFh-80h	Reserved.		
Value	Description										
00h	The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.										
7Fh-01h	When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.										
FFh-80h	Reserved.										
23	<b>Inv: invert counter mask.</b> Read-write. Reset: 0. See CntMask.										
22	<b>En: enable performance counter.</b> Read-write. Reset: 0. 1=Performance event counter is enabled. Performance counter enable.										
21	Reserved.										
20	<b>Int: enable APIC interrupt.</b> Read-write. Reset: 0. 1=APIC performance counter LVT interrupt is enabled to generate an interrupt via Core::X86::Apic::PerformanceCounterLvtEntry when the performance counter overflows. APIC Interrupt enable.										
19	Reserved.										
18	<b>Edge: edge detect.</b> Read-write. Reset: 0. 0=Level detect. 1=Zero-to-one Edge detect. The edge count mode increments the counter when a transition happens on the monitored event. If the event selected is changed without disabling the counter, an extra edge is falsely detected when the first event is a static 0 and the second event is a static one. To avoid this false edge detection, disable the counter when changing the event and then enable the counter with a second MSR write.										
17:16	<b>OsUserMode: OS and user mode.</b> Read-write. Reset: 0h. Event counter for OS and user mode.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Count no events.</td></tr> <tr> <td>1h</td><td>Count user events (CPL&gt;0).</td></tr> <tr> <td>2h</td><td>Count OS events (CPL=0).</td></tr> <tr> <td>3h</td><td>Count all events, irrespective of the CPL.</td></tr> </table>	Value	Description	0h	Count no events.	1h	Count user events (CPL>0).	2h	Count OS events (CPL=0).	3h	Count all events, irrespective of the CPL.
Value	Description										
0h	Count no events.										
1h	Count user events (CPL>0).										
2h	Count OS events (CPL=0).										
3h	Count all events, irrespective of the CPL.										

15:8	<b>UnitMask: event qualification.</b> Read-write. Reset: 00h. Each UnitMask bit further specifies or qualifies the event specified by EventSelect. All events selected by UnitMask are simultaneously monitored. Unless otherwise stated, the UnitMask values shown may be combined (logically ORed) to select any desired combination of the sub-events for a given event. In some cases, certain combinations can result in misleading counts, or the UnitMask value is an ordinal rather than a bit mask. These situations are described where applicable, or should be obvious from the event descriptions. For events where no UnitMask table is shown, the UnitMask is Unused.
7:0	<b>EventSelect[7:0]: event select.</b> Read-write. Reset: 00h. EventSelect[11:0] = {EventSelect[11:8], EventSelect[7:0]}. EventSelect specifies the event or event duration in a processor unit to be counted by the corresponding PERF_CTR[5:0] register. The events are specified in 2.1.14.5 [Core Performance Monitor Counters]. Some events are Reserved; when a Reserved event is selected, the results are undefined.



**MSRC001\_0001 [Performance Event Select 1] (Core::X86::Msr::PERF\_LEGACY\_CTL1)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

The legacy alias of Core::X86::Msr::PERF\_CTL1. See Core::X86::Msr::PERF\_CTL1.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0001

Bits	Description										
63:42	Reserved.										
41:40	<b>HostGuestOnly: count only host/guest events.</b> Read-write. Reset: 0h. Host/Guest event counter.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Count all events, irrespective of guest/host.</td></tr> <tr> <td>1h</td><td>Count guest events if [SVME] == 1.</td></tr> <tr> <td>2h</td><td>Count host events if [SVME] == 1.</td></tr> <tr> <td>3h</td><td>Count all guest and host events if [SVME] == 1.</td></tr> </table>	Value	Description	0h	Count all events, irrespective of guest/host.	1h	Count guest events if [SVME] == 1.	2h	Count host events if [SVME] == 1.	3h	Count all guest and host events if [SVME] == 1.
Value	Description										
0h	Count all events, irrespective of guest/host.										
1h	Count guest events if [SVME] == 1.										
2h	Count host events if [SVME] == 1.										
3h	Count all guest and host events if [SVME] == 1.										
39:36	Reserved.										
35:32	<b>EventSelect[11:8].</b> Read-write. Reset: 0h. Performance event select[11:8].										
31:24	<b>CntMask: counter mask.</b> Read-write. Reset: 00h. Controls the number of events counted per clock cycle.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00h</td><td>The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.</td></tr> <tr> <td>7Fh-01h</td><td>When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.</td></tr> <tr> <td>FFh-80h</td><td>Reserved.</td></tr> </table>	Value	Description	00h	The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.	7Fh-01h	When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.	FFh-80h	Reserved.		
Value	Description										
00h	The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.										
7Fh-01h	When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.										
FFh-80h	Reserved.										
23	<b>Inv: invert counter mask.</b> Read-write. Reset: 0. See CntMask.										
22	<b>En: enable performance counter.</b> Read-write. Reset: 0. 1=Performance event counter is enabled. Performance counter enable.										
21	Reserved.										
20	<b>Int: enable APIC interrupt.</b> Read-write. Reset: 0. 1=APIC performance counter LVT interrupt is enabled to generate an interrupt via Core::X86::Apic::PerformanceCounterLvtEntry when the performance counter overflows. APIC interrupt enable.										
19	Reserved.										
18	<b>Edge: edge detect.</b> Read-write. Reset: 0. 0=Level detect. 1=Zero-to-one Edge detect. The edge count mode increments the counter when a transition happens on the monitored event. If the event selected is changed without disabling the counter, an extra edge is falsely detected when the first event is a static 0 and the second event is a static one. To avoid this false edge detection, disable the counter when changing the event and then enable the counter with a second MSR write.										
17:16	<b>OsUserMode: OS and user mode.</b> Read-write. Reset: 0h. OS and user mode counter events.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Count no events.</td></tr> <tr> <td>1h</td><td>Count user events (CPL&gt;0).</td></tr> <tr> <td>2h</td><td>Count OS events (CPL=0).</td></tr> <tr> <td>3h</td><td>Count all events, irrespective of the CPL.</td></tr> </table>	Value	Description	0h	Count no events.	1h	Count user events (CPL>0).	2h	Count OS events (CPL=0).	3h	Count all events, irrespective of the CPL.
Value	Description										
0h	Count no events.										
1h	Count user events (CPL>0).										
2h	Count OS events (CPL=0).										
3h	Count all events, irrespective of the CPL.										

15:8	<b>UnitMask: event qualification.</b> Read-write. Reset: 00h. Each UnitMask bit further specifies or qualifies the event specified by EventSelect. All events selected by UnitMask are simultaneously monitored. Unless otherwise stated, the UnitMask values shown may be combined (logically ORed) to select any desired combination of the sub-events for a given event. In some cases, certain combinations can result in misleading counts, or the UnitMask value is an ordinal rather than a bit mask. These situations are described where applicable, or should be obvious from the event descriptions. For events where no UnitMask table is shown, the UnitMask is Unused.
7:0	<b>EventSelect[7:0]: event select.</b> Read-write. Reset: 00h. EventSelect[11:0] = {EventSelect[11:8], EventSelect[7:0]}. EventSelect specifies the event or event duration in a processor unit to be counted by the corresponding PERF_CTR[5:0] register. The events are specified in 2.1.14.5 [Core Performance Monitor Counters]. Some events are Reserved; when a Reserved event is selected, the results are undefined.

**MSRC001\_0002 [Performance Event Select 2] (Core::X86::Msr::PERF\_LEGACY\_CTL2)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

The legacy alias of Core::X86::Msr::PERF\_CTL2. See Core::X86::Msr::PERF\_CTL2.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0002

Bits	Description
63:44	Reserved.
43	<b>PreciseRetire.</b> Read-write. Reset: 0. 0=Include events counted during post-retire speculation that were later aborted. 1=Excludes events counted during post-retire speculation that were later aborted.
42	Reserved.
41:40	<b>HostGuestOnly: count only host/guest events.</b> Read-write. Reset: 0h. Host/Guest event counter.
<b>ValidValues:</b>	
Value	Description
0h	Count all events, irrespective of guest/host.
1h	Count guest events if [SVME] == 1.
2h	Count host events if [SVME] == 1.
3h	Count all guest and host events if [SVME] == 1.
39:36	Reserved.
35:32	<b>EventSelect[11:8].</b> Read-write. Reset: 0h. Performance event select[11:8].
31:24	<b>CntMask: counter mask.</b> Read-write. Reset: 00h. Controls the number of events counted per clock cycle.
<b>ValidValues:</b>	
Value	Description
00h	The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.
7Fh-01h	When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.
FFh-80h	Reserved.
23	<b>Inv: invert counter mask.</b> Read-write. Reset: 0. See CntMask.
22	<b>En: enable performance counter.</b> Read-write. Reset: 0. 1=Performance event counter is enabled. Performance counter enable.
21	Reserved.
20	<b>Int: enable APIC interrupt.</b> Read-write. Reset: 0. 1=APIC performance counter LVT interrupt is enabled to generate an interrupt via Core::X86::Apic::PerformanceCounterLvtEntry when the performance counter overflows. APIC interrupt enable.
19	Reserved.
18	<b>Edge: edge detect.</b> Read-write. Reset: 0. 0=Level detect. 1=Zero-to-one Edge detect. The edge count mode increments the counter when a transition happens on the monitored event. If the event selected is changed without disabling the counter, an extra edge is falsely detected when the first event is a static 0 and the second event is a static one. To avoid this false edge detection, disable the counter when changing the event and then enable the counter with a second MSR write.
17:16	<b>OsUserMode: OS and user mode.</b> Read-write. Reset: 0h. OS and user mode.
<b>ValidValues:</b>	
Value	Description
0h	Count no events.
1h	Count user events (CPL>0).
2h	Count OS events (CPL=0).
3h	Count all events, irrespective of the CPL.

15:8	<b>UnitMask: event qualification.</b> Read-write. Reset: 00h. Each UnitMask bit further specifies or qualifies the event specified by EventSelect. All events selected by UnitMask are simultaneously monitored. Unless otherwise stated, the UnitMask values shown may be combined (logically ORed) to select any desired combination of the sub-events for a given event. In some cases, certain combinations can result in misleading counts, or the UnitMask value is an ordinal rather than a bit mask. These situations are described where applicable, or should be obvious from the event descriptions. For events where no UnitMask table is shown, the UnitMask is Unused.
7:0	<b>EventSelect[7:0]: event select.</b> Read-write. Reset: 00h. EventSelect[11:0] = {EventSelect[11:8], EventSelect[7:0]}. EventSelect specifies the event or event duration in a processor unit to be counted by the corresponding PERF_CTR[5:0] register. The events are specified in 2.1.14.5 [Core Performance Monitor Counters]. Some events are Reserved; when a Reserved event is selected, the results are undefined.

**MSRC001\_0003 [Performance Event Select 3] (Core::X86::Msr::PERF\_LEGACY\_CTL3)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

The legacy alias of Core::X86::Msr::PERF\_CTL3. See Core::X86::Msr::PERF\_CTL3.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0003

Bits	Description										
63:42	Reserved.										
41:40	<b>HostGuestOnly: count only host/guest events.</b> Read-write. Reset: 0h. Host/guest event counter.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Count all events, irrespective of guest/host.</td></tr> <tr> <td>1h</td><td>Count guest events if [SVME] == 1.</td></tr> <tr> <td>2h</td><td>Count host events if [SVME] == 1.</td></tr> <tr> <td>3h</td><td>Count all guest and host events if [SVME] == 1.</td></tr> </table>	Value	Description	0h	Count all events, irrespective of guest/host.	1h	Count guest events if [SVME] == 1.	2h	Count host events if [SVME] == 1.	3h	Count all guest and host events if [SVME] == 1.
Value	Description										
0h	Count all events, irrespective of guest/host.										
1h	Count guest events if [SVME] == 1.										
2h	Count host events if [SVME] == 1.										
3h	Count all guest and host events if [SVME] == 1.										
39:36	Reserved.										
35:32	<b>EventSelect[11:8].</b> Read-write. Reset: 0h. Performance event select[11:8].										
31:24	<b>CntMask: counter mask.</b> Read-write. Reset: 00h. Controls the number of events counted per clock cycle.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00h</td><td>The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.</td></tr> <tr> <td>7Fh-01h</td><td>When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.</td></tr> <tr> <td>FFh-80h</td><td>Reserved.</td></tr> </table>	Value	Description	00h	The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.	7Fh-01h	When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.	FFh-80h	Reserved.		
Value	Description										
00h	The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.										
7Fh-01h	When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.										
FFh-80h	Reserved.										
23	<b>Inv: invert counter mask.</b> Read-write. Reset: 0. See CntMask.										
22	<b>En: enable performance counter.</b> Read-write. Reset: 0. 1=Performance event counter is enabled. Performance counter enable.										
21	Reserved.										
20	<b>Int: enable APIC interrupt.</b> Read-write. Reset: 0. 1=APIC performance counter LVT interrupt is enabled to generate an interrupt via Core::X86::Apic::PerformanceCounterLvtEntry when the performance counter overflows. APIC interrupt enable.										
19	Reserved.										
18	<b>Edge: edge detect.</b> Read-write. Reset: 0. 0=Level detect. 1=Zero-to-one Edge detect. The edge count mode increments the counter when a transition happens on the monitored event. If the event selected is changed without disabling the counter, an extra edge is falsely detected when the first event is a static 0 and the second event is a static one. To avoid this false edge detection, disable the counter when changing the event and then enable the counter with a second MSR write.										
17:16	<b>OsUserMode: OS and user mode.</b> Read-write. Reset: 0h. OS and user mode counter events.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Count no events.</td></tr> <tr> <td>1h</td><td>Count user events (CPL&gt;0).</td></tr> <tr> <td>2h</td><td>Count OS events (CPL=0).</td></tr> <tr> <td>3h</td><td>Count all events, irrespective of the CPL.</td></tr> </table>	Value	Description	0h	Count no events.	1h	Count user events (CPL>0).	2h	Count OS events (CPL=0).	3h	Count all events, irrespective of the CPL.
Value	Description										
0h	Count no events.										
1h	Count user events (CPL>0).										
2h	Count OS events (CPL=0).										
3h	Count all events, irrespective of the CPL.										

15:8	<b>UnitMask: event qualification.</b> Read-write. Reset: 00h. Each UnitMask bit further specifies or qualifies the event specified by EventSelect. All events selected by UnitMask are simultaneously monitored. Unless otherwise stated, the UnitMask values shown may be combined (logically ORed) to select any desired combination of the sub-events for a given event. In some cases, certain combinations can result in misleading counts, or the UnitMask value is an ordinal rather than a bit mask. These situations are described where applicable, or should be obvious from the event descriptions. For events where no UnitMask table is shown, the UnitMask is Unused.
7:0	<b>EventSelect[7:0]: event select.</b> Read-write. Reset: 00h. EventSelect[11:0] = {EventSelect[11:8], EventSelect[7:0]}. EventSelect specifies the event or event duration in a processor unit to be counted by the corresponding PERF_CTR[5:0] register. The events are specified in 2.1.14.5 [Core Performance Monitor Counters]. Some events are Reserved; when a Reserved event is selected, the results are undefined.

**MSRC001\_000[4...7] [Performance Event Counter [3:0]] (Core::X86::Msr::PERF\_LEGACY\_CTR)**

Read-write, Volatile. Reset: 0000\_0000\_0000\_0000h.

Note: When counting events that capable of counting greater than 15 events per cycle (MergeEvent) the even and the corresponding odd PERF\_LEGACY\_CTR must be paired to appear as a single 64-bit counter. See 2.1.14.4 [Large Increment per Cycle Events].

The legacy alias of Core::X86::Msr::PERF\_CTR. See Core::X86::Msr::PERF\_CTR.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n0; MSRC001\_0004

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n1; MSRC001\_0005

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n2; MSRC001\_0006

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n3; MSRC001\_0007

Bits	Description
63:48	Reserved.
47:0	<b>CTR.</b> Read-write, Volatile. Reset: 0000_0000_0000h. Performance counter value.

**MSRC001\_0010 [System Configuration] (Core::X86::Msr::SYS\_CFG)**

Reset: 0000\_0000\_0000\_0000h.

If Core::X86::Msr::SYS\_CFG[SecureNestedPagingEn] is set, writes to this register are ignored.

\_lthree[1:0]\_core[7:0]; MSRC001\_0010

Bits	Description															
63:27	Reserved.															
26	<b>HMKEE: Host Multi-Key Encryption Enable.</b> Read,Write-1-only. Reset: 0. Used with SYS_CFG[SMEE] to select secure memory encryption mode. See SYS_CFG[SMEE] for a table listing the available memory encryption modes.															
25	<b>VmplEn.</b> Reset: 0. VM permission levels enable. AccessType: Core::X86::Msr::SYS_CFG[SecureNestedPagingEn] ? Read-only : Read-write.															
24	<b>SecureNestedPagingEn.</b> Read,Error-on-write-1. Reset: 0. Enable Secure Nested Paging (SNP).															
23	<b>SMEE: Secure Memory Encryption Enable.</b> Read,Write-1-only. Reset: 0. <b>Description:</b> Used with SYS_CFG[HMKEE] to select secure memory encryption mode. See the table below for the available memory encryption modes. <table><tr><th>HMKEE</th><th>SMEE</th><th>Description</th></tr><tr><td>0</td><td>0</td><td>No encryption.</td></tr><tr><td>0</td><td>1</td><td>Enables SME and SEV memory encryption.</td></tr><tr><td>1</td><td>0</td><td>Enables SME-HMK memory encryption.</td></tr><tr><td>1</td><td>1</td><td>Not supported. Results in #GP.</td></tr></table>	HMKEE	SMEE	Description	0	0	No encryption.	0	1	Enables SME and SEV memory encryption.	1	0	Enables SME-HMK memory encryption.	1	1	Not supported. Results in #GP.
HMKEE	SMEE	Description														
0	0	No encryption.														
0	1	Enables SME and SEV memory encryption.														
1	0	Enables SME-HMK memory encryption.														
1	1	Not supported. Results in #GP.														
22	<b>Tom2ForceMemTypeWB: top of memory 2 memory type write back.</b> Read-write. Reset: 0. 1=The default memory type of memory between 4GB and Core::X86::Msr::TOM2 is write back instead of the memory type defined by Core::X86::Msr::MTRRdefType[MemType]. For this bit to have any effect, Core::X86::Msr::MTRRdefType[MtrrDefTypeEn] must be 1. MTRRs and PAT can be used to override this memory type.															
21	<b>MtrrTom2En: MTRR top of memory 2 enable.</b> Read-write. Reset: 0. 0=Core::X86::Msr::TOM2 is disabled. 1=Core::X86::Msr::TOM2 is enabled.															
20	<b>MtrrVarDramEn: MTRR variable DRAM enable.</b> Read-write. Reset: 0. Init: BIOS,1. 0=Core::X86::Msr::TOP_MEM and IORRs are disabled. 1=These registers are enabled.															
19	<b>MtrrFixDramModEn: MTRR fixed RdDram and WrDram modification enable.</b> Read-write. Reset: 0. 0=Core::X86::Msr::MtrrFix_64K through Core::X86::Msr::MtrrFix_4K_7 [RdDram,WrDram] read values is masked 00b; writing does not change the hidden value. 1=Core::X86::Msr::MtrrFix_64K through Core::X86::Msr::MtrrFix_4K_7 [RdDram,WrDram] access type is Read-write. Not shared between threads. Controls access to Core::X86::Msr::MtrrFix_64K through Core::X86::Msr::MtrrFix_4K_7 [RdDram ,WrDram]. This bit should be set to 1 during BIOS initialization of the fixed MTRRs, then cleared to 0 for operation.															
18	<b>MtrrFixDramEn: MTRR fixed RdDram and WrDram attributes enable.</b> Read-write. Reset: 0. Init: BIOS,1. 1=Enables the RdDram and WrDram attributes in Core::X86::Msr::MtrrFix_64K through Core::X86::Msr::MtrrFix_4K_7.															
17:0	Reserved.															

**MSRC001\_0015 [Hardware Configuration] (Core::X86::Msr::HWCR)**

Reset: 0000\_0000\_0100\_6010h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0015

Bits	Description
63:36	Reserved.
35	<b>CpuidFltEn.</b> Read-write. Reset: 0. 1=Executing CPUID outside of SMM and with CPL > 0 results in #GP.
34	<b>DownGradeFp512ToFP256.</b> Read-write. Reset: 0. 1=Downgrade FP512 performance to look more like FP256 performance.
33	<b>SmmPgCfgLock.</b> Read-write. Reset: 0. 1=SMM page config locked. Error-on-write-1 if not in SMM mode. RSM unconditionally clears Core::X86::Msr::HWCR[SmmPgCfgLock].
32:31	Reserved.
30	<b>IRPerfEn: enable instructions retired counter.</b> Read-write. Reset: 0. 1=Enable Core::X86::Msr::IRPerfCount.
29:28	Reserved.
27	<b>EffFreqReadOnlyLock: read-only effective frequency counter lock.</b> Write-1-only. Reset: 0. Init: BIOS, 1. 1=Core::X86::Msr::MPerfReadOnly, Core::X86::Msr::APerfReadOnly and Core::X86::Msr::IRPerfCount are read-only.
26	<b>EffFreqCntMwait: effective frequency counting during mwait.</b> Read-write. Reset: 0. 0=The registers do not increment. 1=The registers increment. Specifies whether Core::X86::Msr::MPERF and Core::X86::Msr::APERF increment while the core is in the monitor event pending state. See 2.1.4 [Effective Frequency].
25	<b>CpbDis: core performance boost disable.</b> Read-write. Reset: 0. 0=CPB is requested to be enabled. 1=CPB is disabled. Specifies whether core performance boost is requested to be enabled or disabled. If core performance boost is disabled while a core is in a boosted P-state, the core automatically transitions to the highest performance non-boosted P-state.
24	<b>TscFreqSel: TSC frequency select.</b> Read-only. Reset: 1. 1=The TSC increments at the P0 frequency.
23:22	Reserved.
21	<b>LockTscToCurrentP0: lock the TSC to the current P0 frequency.</b> Read-write. Reset: 0. 0=The TSC will count at the P0 frequency. 1=The TSC frequency is locked to the current P0 frequency at the time this bit is set and remains fixed regardless of future changes to the P0 frequency.
20	<b>IoCfgGpFault: IO-space configuration causes a GP fault.</b> Read-write. Reset: 0. 1=IO-space accesses to configuration space cause a GP fault. The fault is triggered if any part of the IO Read/Write address range is between CF8h and CFFh, inclusive. These faults only result from single IO instructions, not to string and REP IO instructions. This fault takes priority over the IO trap mechanism described by Core::X86::Msr::SMI_ON_IO_TRAP_CTL_STS.
19	Reserved.
18	<b>McStatusWrEn: machine check status write enable.</b> Read-write. Reset: 0. 0=MCA_STATUS registers are readable; writing a non-zero pattern to these registers causes a general protection fault. 1=MCA_STATUS registers are Read-write, including Reserved fields; do not cause general protection faults; such writes update all implemented bits in these registers; All fields of all threshold registers are Read-write when accessed from MSR space, including Locked, except BlkPtr which is always Read-only; McStatusWrEn does not change the access type for the thresholding registers accessed via configuration space. <b>Description:</b> McStatusWrEn can be used to debug machine check exception and interrupt handlers. Independent of the value of this bit, the processor may enforce Write-Ignored behavior on MCA_STATUS registers depending on platform settings. See 3.1 [Machine Check Architecture].
17	<b>Wrap32Dis: 32-bit address wrap disable.</b> Read-write. Reset: 0. 1=Disable 32-bit address wrapping. Software can use Wrap32Dis to access physical memory above 4 Gbytes without switching into 64-bit mode. To do so, software should write a greater-than 4 Gbyte address to Core::X86::Msr::FS_BASE and Core::X86::Msr::GS_BASE. Then it would address $\pm 2$ Gbytes from one of those bases using normal memory reference instructions with a FS or GS override prefix. However, the INVLPG, FST, and SSE store instructions generate 32-bit addresses in legacy mode, regardless of the state of Wrap32Dis.
16:15	Reserved.



14	<b>RsmSpCycDis: RSM special bus cycle disable.</b> Reset: 1. Init: BIOS,1. 0=A link special bus cycle, SMIACK, is generated on a resume from SMI. AccessType: Core::X86::Msrr::HWCR[SmmLock] ? Read-only : Read-write.
13	<b>SmiSpCycDis: SMI special bus cycle disable.</b> Reset: 1. Init: BIOS,1. 0=A link special bus cycle, SMIACK, is generated when an SMI interrupt is taken. AccessType: Core::X86::Msrr::HWCR[SmmLock] ? Read-only : Read-write.
12:11	Reserved.
10	<b>MonMwaitUserEn: MONITOR/MWAIT user mode enable.</b> Read-write. Reset: 0. 0=The MONITOR and MWAIT instructions are supported only in privilege level 0; these instructions in privilege levels 1 to 3 cause a #UD exception. 1=The MONITOR and MWAIT instructions are supported in all privilege levels. The state of this bit is ignored if MonMwaitDis is set.
9	<b>MonMwaitDis: MONITOR and MWAIT disable.</b> Read-write. Reset: 0. 1=The MONITOR, MWAIT, MONITORX, and MWAITX opcodes become invalid. This affects what is reported back through Core::X86::Cpuid::FeatureIdEcxC[Monitor] and Core::X86::Cpuid::FeatureExtIdEcxC[MwaitExtended].
8	<b>IgnneEm: IGNNE port emulation enable.</b> Read-write. Reset: 0. 1=Enable emulation of IGNNE port.
7	<b>AllowFERRonNE: allow FERR on NE.</b> Read-write. Reset: 0. 0=Disable FERR signalling when generating an x87 floating point exception (when CR0[NE] is set). 1=FERR is signaled on any x87 floating point exception, regardless of CR0[NE].
6:5	Reserved.
4	<b>INVDWBINVD: INVD to WBINVD conversion.</b> Read,Error-on-write-0. Reset: 1. 1=Convert INVD to WBINVD. <b>Description:</b> This bit is required to be set for normal operation when any of the following are true: <ul style="list-style-type: none"> <li>• An L2 is shared by multiple threads.</li> <li>• An L3 is shared by multiple cores.</li> <li>• CC6 is enabled.</li> <li>• Probe filter is enabled.</li> </ul>
3	<b>TlbCacheDis: cacheable memory disable.</b> Read-write. Reset: 0. 1=Disable performance improvement that assumes that the PML4, PDP, PDE and PTE entries are in cacheable WB DRAM. <b>Description:</b> Operating systems that maintain page tables in any other memory type must set the TlbCacheDis bit to insure proper operation. Operating system should do a full TLB flush before and after any changes to this bit value. <ul style="list-style-type: none"> <li>• TlbCacheDis does not override the memory type specified by the SMM ASeg and TSeg memory regions controlled by Core::X86::Msrr::SMMAddr Core::X86::Msrr::SMMMMask.</li> </ul>
2:1	Reserved.
0	<b>SmmLock: SMM code lock.</b> Read,Write-1-only. Reset: 0. Init: BIOS,1. 1=SMM code in the ASeg and TSeg range and the SMM registers are Read-only and SMI interrupts are not intercepted in SVM. See 2.1.11.1.10 [Locking SMM].

**MSRC001\_001[6...8] [IO Range Base] (Core::X86::Msr::IORR\_BASE)**

Read-write.

Core::X86::Msr::IORR\_BASE and Core::X86::Msr::IORR\_MASK combine to specify the two sets of base and mask pairs for two IORR ranges. A core access, with address CPUAddr, is determined to be within IORR address range if the following equation is true:

$\text{CPUAddr}[47:12] \& \text{PhyMask}[47:12] == \text{PhyBase}[47:12] \& \text{PhyMask}[47:12]$ .

BIOS can use the IORRs to create an IO hole within a range of addresses that would normally be mapped to DRAM. It can also use the IORRs to re-assert a DRAM destination for a range of addresses that fall within a bigger IO hole that overlays DRAM.

If Core::X86::Msr::SYS\_CFG[SecureNestedPagingEn] is set to 1, this register will be read-only and attempts to write to this register will result in #GP(0).

\_lthree[1:0]\_core[7:0]\_n0; MSRC001\_0016

\_lthree[1:0]\_core[7:0]\_n1; MSRC001\_0018

Bits	Description
63:48	Reserved.
47:12	<b>PhyBase.</b> Read-write. Reset: X_XXXX_XXXXh. Physical base address for IO range.
11:5	Reserved.
4	<b>RdMem: read from memory.</b> Read-write. Reset: X. 0=Read accesses to the range are directed to IO. 1=Read accesses to the range are directed to system memory.
3	<b>WrMem: write to memory.</b> Read-write. Reset: X. 0=Write accesses to the range are directed to IO. 1=Write accesses to the range are directed to system memory.
2:0	Reserved.

**MSRC001\_001[7...9] [IO Range Mask] (Core::X86::Msr::IORR\_MASK)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

See Core::X86::Msr::IORR\_BASE.

If Core::X86::Msr::SYS\_CFG[SecureNestedPagingEn] is set to 1, this register will be read-only and attempts to write to this register will result in #GP(0).

\_lthree[1:0]\_core[7:0]\_n0; MSRC001\_0017

\_lthree[1:0]\_core[7:0]\_n1; MSRC001\_0019

Bits	Description
63:48	Reserved.
47:12	<b>PhyMask.</b> Read-write. Reset: 0_0000_0000h. Physical address mask for IO range.
11	<b>Valid.</b> Read-write. Reset: 0. 1=The pair of registers that specifies an IORR range is valid.
10:0	Reserved.

**MSRC001\_001A [Top Of Memory] (Core::X86::Msr::TOP\_MEM)**

Read-write.

If Core::X86::Msr::SYS\_CFG[SecureNestedPagingEn] is set to 1, this register will be read-only and attempts to write to this register will result in #GP(0).

\_lthree[1:0]\_core[7:0]; MSRC001\_001A

Bits	Description
63:48	Reserved.
47:23	<b>TOM[47:23]: top of memory.</b> Read-write. Reset: XXX_XXXXh. Specifies the address that divides between MMIO and DRAM. This value is normally placed below 4G. From TOM to 4G is MMIO; below TOM is DRAM. See 2.1.5.3 [System Address Map].
22:0	Reserved.

**MSRC001\_001D [Top Of Memory 2] (Core::X86::Msr::TOM2)**

Read-write.

If Core::X86::Msr::SYS\_CFG[SecureNestedPagingEn] is set to 1, this register will be read-only and attempts to write to this register will result in #GP(0).

\_lthree[1:0]\_core[7:0]; MSRC001\_001D

Bits	Description
63:48	Reserved.
47:23	<b>TOM2[47:23]: second top of memory.</b> Read-write. Reset: XXX_XXXXh. Specifies the address divides between MMIO and DRAM. This value is normally placed above 4G. From 4G to TOM2 - 1 is DRAM; TOM2 and above is MMIO. See 2.1.5.3 [System Address Map]. This register is enabled by Core::X86::Msr::SYS_CFG[MtrrTom2En].
22:0	Reserved.

**MSRC001\_0020 [Patch Loader] (Core::X86::Msr::PATCH\_LOADER)**

Write-only, Error-on-read.

\_lthree[1:0]\_core[7:0]; MSRC001\_0020

Bits	Description
63:0	<b>PatchBase.</b> Write-only, Error-on-read. Reset: XXXX_XXXX_XXXX_XXXXh. Linear address of the Microcode Patch Block. PatchBase[63:32] is ignored when the core is not operating in 64-bit mode.

**MSRC001\_0022 [Machine Check Exception Redirection] (Core::X86::Msr::McExcepRedir)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

This register can be used to redirect machine check exceptions (MCEs) to SMIs or vectored interrupts. If both RedirSmiEn and RedirVecEn are set, then undefined behavior results.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0022

Bits	Description
63:10	Reserved.
9	<b>RedirSmiEn.</b> Read-write. Reset: 0. 1=Redirect MCEs (that are directed to this core) to generate an SMI-trigger IO cycle via Core::X86::Msr::SmiTrigIoCycle. The status is stored in Core::X86::Smm::LocalSmiStatus[MceRedirSts].
8	<b>RedirVecEn.</b> Read-write. Reset: 0. 1=Redirect MCEs (that are directed to this core) to generate a vectored interrupt, using the interrupt vector specified in RedirVector.
7:0	<b>RedirVector.</b> Read-write. Reset: 00h. See RedirVecEn.

**MSRC001\_003[0...5] [Processor Name String] (Core::X86::Msr::ProcNameString)**

Read-write.

These 6 registers hold the CPUID name string in ASCII. The state of these registers are returned by CPUID instructions, Core::X86::Cpuid::ProcNameStr0Eax through Core::X86::Cpuid::ProcNameStr2Edx. BIOS should set these registers to the product name for the processor as provided by AMD. Each register contains a block of 8 ASCII characters; the least byte corresponds to the first ASCII character of the block; the most-significant byte corresponds to the last character of the block. MSRC001\_0030 contains the first block of the name string; MSRC001\_0035 contains the last block of the name string.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n0; MSRC001\_0030

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n1; MSRC001\_0031

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n2; MSRC001\_0032

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n3; MSRC001\_0033

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n4; MSRC001\_0034

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n5; MSRC001\_0035

Bits	Description
63:56	<b>CpuNameString7</b> . Read-write. Reset: XXh. CPUID name string in ASCII.
55:48	<b>CpuNameString6</b> . Read-write. Reset: XXh. CPUID name string in ASCII.
47:40	<b>CpuNameString5</b> . Read-write. Reset: XXh. CPUID name string in ASCII.
39:32	<b>CpuNameString4</b> . Read-write. Reset: XXh. CPUID name string in ASCII.
31:24	<b>CpuNameString3</b> . Read-write. Reset: XXh. CPUID name string in ASCII.
23:16	<b>CpuNameString2</b> . Read-write. Reset: XXh. CPUID name string in ASCII.
15:8	<b>CpuNameString1</b> . Read-write. Reset: XXh. CPUID name string in ASCII.
7:0	<b>CpuNameString0</b> . Read-write. Reset: XXh. CPUID name string in ASCII.

**MSRC001\_005[0...3] [IO Trap] (Core::X86::Msr::SMI\_ON\_IO\_TRAP)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

Core::X86::Msr::SMI\_ON\_IO\_TRAP and Core::X86::Msr::SMI\_ON\_IO\_TRAP\_CTL\_STS provide a mechanism for executing the SMI handler if an access to one of the specified addresses is detected. Access address and access type checking is performed before IO instruction execution. If the access address and access type match one of the specified IO address and access types, then: (1) the IO instruction is not executed; (2) any breakpoint, other than the single-step breakpoint, set on the IO instruction is not taken (the single-step breakpoint is taken after resuming from SMM); and (3) issue the SMI-trigger IO cycle specified by Core::X86::Msr::SmiTrigIoCycle if enabled. The status is stored in Core::X86::Smm::LocalSmiStatus[IoTrapSts].

IO-space configuration accesses are special IO accesses. An IO access is defined as an IO-space configuration access when IO instruction address bits[31:0] are CFCh, CFDh, CFEh, or CFFh when IO-space configuration is enabled (IO::IoCfgAddr[ConfigEn]). The access address for a configuration space access is the current value of IO::IoCfgAddr[BusNo,Device,Function,RegNo]. The access address for an IO access that is not a configuration access is equivalent to the IO instruction address, bits[31:0].

The access address is compared with SmiAddr, and the instruction access type is compared with the enabled access types defined by ConfigSmi, SmiOnRdEn, and SmiOnWrEn. Access address bits[23:0] can be masked with SmiMask. IO and configuration space trapping to SMI applies only to single IO instructions; it does not apply to string and REP IO instructions. The conditional GP fault described by Core::X86::Msr::HWCR[IoCfgGpFault] takes priority over this trap.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n0; MSRC001\_0050

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n1; MSRC001\_0051

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n2; MSRC001\_0052

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n3; MSRC001\_0053

Bits	Description
63	<b>SmiOnRdEn: enable SMI on IO Read.</b> Read-write. Reset: 0. 1=Enables SMI generation on a Read access.
62	<b>SmiOnWrEn: enable SMI on IO Write.</b> Read-write. Reset: 0. 1=Enables SMI generation on a Write access.
61	<b>ConfigSmi: configuration space SMI.</b> Read-write. Reset: 0. 0=IO access (that is not an IO-space configuration access). 1=Configuration access.
60:56	Reserved.
55:32	<b>SmiMask[23:0].</b> Read-write. Reset: 00_0000h. SMI IO trap mask.
<b>ValidValues:</b>	
Value	Description
0	Mask address bit
1	Do not mask address bit
31:0	<b>SmiAddr[31:0].</b> Read-write. Reset: 0000_0000h. SMI IO trap address.

**MSRC001\_0054 [IO Trap Control] (Core::X86::Msr::SMI\_ON\_IO\_TRAP\_CTL\_STS)**

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0054

Bits	Description
63:16	Reserved.
15	<b>IoTrapEn: IO trap enable.</b> Read-write. Reset: 0. 1=Enable IO and configuration space trapping specified by Core::X86::Msr::SMI_ON_IO_TRAP and Core::X86::Msr::SMI_ON_IO_TRAP_CTL_STS.
14:8	Reserved.
7	<b>SmiEn3.</b> Read-write. Reset: 0. 1=The trap Core::X86::Msr::SMI_ON_IO_TRAP_n[3] is enabled.
6	Reserved.
5	<b>SmiEn2.</b> Read-write. Reset: 0. 1=The trap Core::X86::Msr::SMI_ON_IO_TRAP_n[2] is enabled.
4	Reserved.
3	<b>SmiEn1.</b> Read-write. Reset: 0. 1=The trap Core::X86::Msr::SMI_ON_IO_TRAP_n[1] is enabled.
2	Reserved.
1	<b>SmiEn0.</b> Read-write. Reset: 0. 1=The trap Core::X86::Msr::SMI_ON_IO_TRAP_n[0] is enabled.
0	Reserved.

**MSRC001\_0055 [Reserved.] (Core::X86::Msr::IntPend)**

Read-only. Reset: Fixed, 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]; MSRC001\_0055

**Bits Description**

63:0 Reserved.

**MSRC001\_0056 [SMI Trigger IO Cycle] (Core::X86::Msr::SmiTrigIoCycle)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

See 2.1.11.1.3 [SMI Sources And Delivery]. This register specifies an IO cycle that may be generated when a local SMI trigger event occurs. If IoCycleEn is set and there is a local SMI trigger event, then the IO cycle generated is a byte Read or Write, based on IoRd, to address IoPortAddress. If the cycle is a Write, then IoData contains the data written. If the cycle is a Read, the value read is discarded. If IoCycleEn is clear and a local SMI trigger event occurs, then undefined behavior results.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0056

**Bits Description**

63:27 Reserved.

26 **IoRd: IO Read.** Read-write. Reset: 0. 0=IO write. 1=IO read.25 **IoCycleEn: IO cycle enable.** Read-write. Reset: 0. 1=The SMI trigger IO cycle is enabled to be generated.

24 Reserved.

23:16 **IoData.** Read-write. Reset: 00h. See 2.1.11.1.3 [SMI Sources And Delivery].15:0 **IoPortAddress.** Read-write. Reset: 0000h. See 2.1.11.1.3 [SMI Sources And Delivery].

**MSRC001\_0058 [MMIO Configuration Base Address] (Core::X86::Msr::MmioCfgBaseAddr)**

See 2.1.6 [Configuration Space] for a description of MMIO configuration space.

\_lthree[1:0]\_core[7:0]; MSRC001\_0058

Bits	Description																																		
63:48	Reserved.																																		
47:20	<b>MmioCfgBaseAddr[47:20]: MMIO configuration base address bits[47:20].</b> Read-write. Reset: XXX_XXXXh. Specifies the base address of the MMIO configuration range.																																		
19:6	Reserved.																																		
5:2	<b>BusRange: bus range identifier.</b> Read-write. Reset: 0h. Specifies the number of buses in the MMIO configuration space range. The size of the MMIO configuration space is 1 MB times the number of buses. <b>ValidValues:</b>																																		
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr><td>0h</td><td>1 bus, 1 segment</td></tr> <tr><td>1h</td><td>2 buses, 1 segment</td></tr> <tr><td>2h</td><td>4 buses, 1 segment</td></tr> <tr><td>3h</td><td>8 buses, 1 segment</td></tr> <tr><td>4h</td><td>16 buses, 1 segment</td></tr> <tr><td>5h</td><td>32 buses, 1 segment</td></tr> <tr><td>6h</td><td>64 buses, 1 segment</td></tr> <tr><td>7h</td><td>128 buses, 1 segment</td></tr> <tr><td>8h</td><td>256 buses, 1 segment</td></tr> <tr><td>9h</td><td>2 segments, 256 buses per segment</td></tr> <tr><td>Ah</td><td>4 segments, 256 buses per segment</td></tr> <tr><td>Bh</td><td>8 segments, 256 buses per segment</td></tr> <tr><td>Ch</td><td>16 segments, 256 buses per segment</td></tr> <tr><td>Dh</td><td>32 segments, 256 buses per segment</td></tr> <tr><td>Eh</td><td>64 segments, 256 buses per segment</td></tr> <tr><td>Fh</td><td>128 segments, 256 buses per segment</td></tr> </table>	Value	Description	0h	1 bus, 1 segment	1h	2 buses, 1 segment	2h	4 buses, 1 segment	3h	8 buses, 1 segment	4h	16 buses, 1 segment	5h	32 buses, 1 segment	6h	64 buses, 1 segment	7h	128 buses, 1 segment	8h	256 buses, 1 segment	9h	2 segments, 256 buses per segment	Ah	4 segments, 256 buses per segment	Bh	8 segments, 256 buses per segment	Ch	16 segments, 256 buses per segment	Dh	32 segments, 256 buses per segment	Eh	64 segments, 256 buses per segment	Fh	128 segments, 256 buses per segment
Value	Description																																		
0h	1 bus, 1 segment																																		
1h	2 buses, 1 segment																																		
2h	4 buses, 1 segment																																		
3h	8 buses, 1 segment																																		
4h	16 buses, 1 segment																																		
5h	32 buses, 1 segment																																		
6h	64 buses, 1 segment																																		
7h	128 buses, 1 segment																																		
8h	256 buses, 1 segment																																		
9h	2 segments, 256 buses per segment																																		
Ah	4 segments, 256 buses per segment																																		
Bh	8 segments, 256 buses per segment																																		
Ch	16 segments, 256 buses per segment																																		
Dh	32 segments, 256 buses per segment																																		
Eh	64 segments, 256 buses per segment																																		
Fh	128 segments, 256 buses per segment																																		
1	Reserved.																																		
0	<b>Enable.</b> Read-write. Reset: 0. 1=MMIO configuration space is enabled.																																		

**MSRC001\_0061 [P-state Current Limit] (Core::X86::Msr::PStateCurLim)**

\_lthree[1:0]\_core[7:0]; MSRC001\_0061

Bits	Description
63:7	Reserved.
6:4	<b>PstateMaxVal: P-state maximum value.</b> Read,Error-on-write,Volatile. Reset: XXXb. Specifies the lowest-performance non-boosted P-state (highest non-boosted value) allowed. Attempts to change Core::X86::Msr::PStateCtl[PstateCmd] to a lower-performance P-state (higher value) are clipped to the value of this field.
3	Reserved.
2:0	<b>CurPstateLimit: current P-state limit.</b> Read,Error-on-write,Volatile. Reset: XXXb. Specifies the highest-performance P-state (lowest value) allowed. CurPstateLimit is always bounded by Core::X86::Msr::PStateCurLim[PstateMaxVal]. Attempts to change the CurPstateLimit to a value greater (lower performance) than Core::X86::Msr::PStateCurLim[PstateMaxVal] leaves CurPstateLimit unchanged.

**MSRC001\_0062 [P-state Control] (Core::X86::Msr::PStateCtl)**

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0062

Bits	Description
63:3	Reserved.
2:0	<b>PstateCmd: P-state change command.</b> Read-write. Reset: XXXb. Cold reset value varies by product; after a warm reset, value initializes to the P-state the core was in prior to the reset. Writes to this field cause the core to change to the indicated non-boosted P-state number, specified by Core::X86::Msr::PStateDef. 0=P0, 1=P1, etc. P-state limits are applied to any P-state requests made through this register. Reads from this field return the last written value, regardless of whether any limits are applied.

**MSRC001\_0063 [P-state Status] (Core::X86::Msr::PStateStat)**

Read,Error-on-write, Volatile.

\_lthree[1:0]\_core[7:0]; MSRC001\_0063

Bits	Description
63:3	Reserved.
2:0	<b>CurPstate: current P-state.</b> Read,Error-on-write, Volatile. Reset: XXXb. This field provides the frequency component of the current non-boosted P-state of the core (regardless of the source of the P-state change, including Core::X86::Msr::PStateCtl[PstateCmd]. 0=P0, 1=P1, etc.). The value of this field is updated when the COF transitions to a new value associated with a P-state.



**MSRC001\_006[4...B] [P-state [7:0]] (Core::X86::Msr::PStateDef)**

Read-write.

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

The CpuVid field in these registers is required to be programmed to the same value in all cores of a processor, but are allowed to be different between processors in a multi-processor system. All other fields in these registers are required to be programmed to the same value in each core of the coherent fabric.

\_n0; MSRC001\_0064

\_n1; MSRC001\_0065

\_n2; MSRC001\_0066

\_n3; MSRC001\_0067

\_n4; MSRC001\_0068

\_n5; MSRC001\_0069

\_n6; MSRC001\_006A

\_n7; MSRC001\_006B

Bits	Description
63	<b>PstateEn.</b> Read-write. Reset: X. 0=The P-state specified by this MSR is not valid. 1=The P-state specified by this MSR is valid. The purpose of this register is to indicate if the rest of the P-state information in the register is valid after a reset; it controls no hardware.
62:33	Reserved.
32	<b>CpuVid[8]: core VID[8].</b> Read-write. Reset: X.
31:30	<b>IddDiv: current divisor.</b> Read-write. Reset: XXb. See IddValue.
29:22	<b>IddValue: current value.</b> Read-write. Reset: XXXXXXXXb. After a reset, IddDiv and IddValue combine to specify the expected maximum current dissipation of a single core that is in the P-state corresponding to the MSR number. These values are intended to be used to create ACPI-defined _PSS objects. The values are expressed in amps; they are not intended to convey final product power levels; they may not match the power levels specified in the Power and Thermal Datasheets.
21:14	<b>CpuVid[7:0]: core VID[7:0].</b> Read-write. Reset: XXXXXXXXb.
13:12	Reserved.
11:0	<b>CpuFid[11:0]: core frequency ID.</b> Read-write. Reset: XXXh. Specifies the core frequency multiplier. The core COF is a function of CpuFid and CpuDid, and defined by CoreCOF.
<b>ValidValues:</b>	
<b>Value</b>	<b>Description</b>
00Fh-000h	Reserved.
FFFh-010h	<Value>*5

**MSRC001\_0073 [C-state Base Address] (Core::X86::Msr::CStateBaseAddr)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0073

Bits	Description
63:16	Reserved.
15:0	<b>CstateAddr: C-state address.</b> Read-write. Reset: 0000h. Specifies the IO addresses trapped by the core for C-state entry requests. A value of 0 in this field specifies that the core does not trap any IO addresses for C-state entry. Writing values greater than FFF8h into this field result in undefined behavior. All other values cause the core to trap IO addresses CstateAddr through CstateAddr + 7.

**MSRC001\_0111 [SMM Base Address] (Core::X86::Msrr::SMM\_BASE)**

Reset: 0000\_0000\_0003\_0000h.

This holds the base of the SMM memory region. The value of this register is stored in the save state on entry into SMM (see 2.1.11.1.5 [SMM Save State]) and it is restored on returning from SMM. The 16-bit CS (code segment) selector is loaded with SmmBase[19:4] on entering SMM. SmmBase[3:0] is required to be 0. The SMM base address can be changed in two ways:

- The SMM base address, at offset FF00h in the SMM state save area, may be changed by the SMI handler. The RSM instruction updates SmmBase with the new value.
- Normal WRMSR access to this register.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0111

Bits	Description
63:32	Reserved.
31:0	<b>SmmBase.</b> Reset: 0003_0000h. Base address of the SMM memory region. AccessType: Core::X86::Msrr::HWCR[SmmLock] ? Read-only : Read-write.

**MSRC001\_0112 [SMM TSeg Base Address] (Core::X86::Msrr::SMMAddr)**

Configurable. Reset: 0000\_0000\_0000\_0000h.

See 2.1.11.1 [System Management Mode (SMM)] and 2.1.5.3.1 [Memory Access to the Physical Address Space]. See Core::X86::Msrr::SMMMask for more information about the ASeg and TSeg address ranges.

Each CPU access, directed at CPUAddr, is determined to be in the TSeg range if the following is true:

$\text{CPUAddr}[47:17] \& \text{TSegMask}[47:17] == \text{TSegBase}[47:17] \& \text{TSegMask}[47:17]$ .

For example, if TSeg spans 256 KB and starts at the 1 MB address. The Core::X86::Msrr::SMMAddr[TSegBase[47:17]] would be set to 0010\_0000h and the Core::X86::Msrr::SMMMask[TSegMask[47:17]] to FFFC\_0000h (with zeros filling in for bits[16:0]). This results in a TSeg range from 0010\_0000 to 0013\_FFFFh.

\_lthree[1:0]\_core[7:0]; MSRC001\_0112

Bits	Description
63:48	Reserved.
47:17	<b>TSegBase[47:17]: TSeg address range base.</b> Configurable. Reset: 0000_0000h. AccessType: (Core::X86::Msrr::HWCR[SmmLock]) ? Read-only : Read-write.
16:0	Reserved.

**MSRC001\_0113 [SMM TSeg Mask] (Core::X86::Msrr::SMMMask)**

Configurable. Reset: 0000\_0000\_0000\_0000h.

See 2.1.11.1 [System Management Mode (SMM)].

The ASeg address range is located at a fixed address from A0000h–BFFFFh. The TSeg range is located at a variable base (specified by Core::X86::Msrr::SMMAddr[TSegBase[47:17]]) with a variable size (specified by Core::X86::Msrr::SMMMask[TSegMask[47:17]]). These ranges provide a safe location for SMM code and data that is not readily accessible by non-SMM applications. The SMI handler can be located in one of these two ranges, or it can be located outside these ranges. These ranges must never overlap each other.

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

- If [A,T]Valid == 1, then:
  - If in SMM, then:
    - If [A,T]Close == 0, then the accesses are directed to DRAM with memory type as specified in [A,T]MTypeDram.
    - If [A,T]Close == 1, then instruction accesses are directed to DRAM with memory type as specified in [A,T]MTypeDram and data accesses are directed at MMIO space and with attributes based on [A,T]MTypeIoWc.
  - If not in SMM, then the accesses are directed at MMIO space with attributes based on [A,T]MTypeIoWc.
- See 2.1.5.3.1.1 [Determining Memory Type].

\_lthree[1:0]\_core[7:0]; MSRC001\_0113

Bits	Description
63:48	Reserved.
47:17	<b>TSegMask[47:17]: TSeg address range mask.</b> Configurable. Reset: 0000_0000h. See Core::X86::Msrr::SMMAddr. AccessType: (Core::X86::Msrr::HWCR[SmmLock]) ? Read-only : Read-write.
16:15	Reserved.
14:12	<b>TMTypeDram: TSeg address range memory type.</b> Configurable. Reset: 0h. Specifies the memory type for SMM accesses to the TSeg range that are directed to DRAM. AccessType: (Core::X86::Msrr::HWCR[SmmLock]) ? Read-only : Read-write.
<b>ValidValues:</b>	
Value	Description
0h	UC or uncacheable.
1h	WC or write combining.
3h-2h	Reserved.
4h	WT or write through.
5h	WP or write protect.
6h	WB or write back.
7h	Reserved.
11	Reserved.
10:8	<b>AMTypeDram: ASeg Range Memory Type.</b> Configurable. Reset: 0h. Specifies the memory type for SMM accesses to the ASeg range that are directed to DRAM. AccessType: (Core::X86::Msrr::HWCR[SmmLock]) ? Read-only : Read-write.

<b>ValidValues:</b>	
<b>Value</b>	<b>Description</b>
0h	UC or uncacheable.
1h	WC or write combining.
3h-2h	Reserved.
4h	WT or write through.
5h	WP or write protect.
6h	WB or write back.
7h	Reserved.
7:6	Reserved.
5	<b>TMTypeIoWc: non-SMM TSeg address range memory type.</b> Configurable. Reset: 0. 0=UC (uncacheable). 1=WC (write combining). Specifies the attribute of TSeg accesses that are directed to MMIO space. AccessType: (Core::X86::Msr::HWCR[SmmLock]) ? Read-only : Read-write.
4	<b>AMTypeIoWc: non-SMM ASeg address range memory type.</b> Configurable. Reset: 0. 0=UC (uncacheable). 1=WC (write combining). Specifies the attribute of ASeg accesses that are directed to MMIO space. AccessType: (Core::X86::Msr::HWCR[SmmLock]) ? Read-only : Read-write.
3	<b>TClose: send TSeg address range data accesses to MMIO.</b> Configurable. Reset: 0. 1=When in SMM, direct data accesses in the TSeg address range to MMIO space. See AClose. AccessType: (Core::X86::Msr::HWCR[SmmLock]) ? Read-only : Read-write.
2	<b>AClose: send ASeg address range data accesses to MMIO.</b> Configurable. Reset: 0. 1=When in SMM, direct data accesses in the ASeg address range to MMIO space. [A,T]Close allows the SMI handler to access the MMIO space located in the same address region as the [A,T]Seg. When the SMI handler is finished accessing the MMIO space, it must clear the bit. Failure to do so before resuming from SMM causes the CPU to erroneously read the save state from MMIO space. AccessType: (Core::X86::Msr::HWCR[SmmLock]) ? Read-only : Read-write.
1	<b>TValid: enable TSeg SMM address range.</b> Configurable. Reset: 0. 1=The TSeg address range SMM enabled. AccessType: (Core::X86::Msr::HWCR[SmmLock]) ? Read-only : Read-write.
0	<b>AValid: enable ASeg SMM address range.</b> Configurable. Reset: 0. 1=The ASeg address range SMM enabled. AccessType: (Core::X86::Msr::HWCR[SmmLock]) ? Read-only : Read-write.

#### MSRC001\_0114 [Virtual Machine Control] (Core::X86::Msr::VM\_CR)

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0114

<b>Bits</b>	<b>Description</b>
63:5	Reserved.
4	<b>SvmeDisable: SVM disable.</b> Configurable. Reset: 0. 0=Core::X86::Msr::EFER[SVME] is Read-write. 1=Core::X86::Msr::EFER[SVME] is Read-only,Error-on-write-1. See Lock for the access type of this field. Attempting to set this field when (Core::X86::Msr::EFER[SVME] == 1) causes a #GP fault, regardless of the state of Lock. See the docAPM2 section titled "Enabling SVM" for software use of this field.
3	<b>Lock: SVM lock.</b> Read-only,Volatile. Reset: 0. 0=SvmeDisable is Read-write. 1=SvmeDisable is Read-only. See Core::X86::Msr::SvmLockKey[SvmLockKey] for the condition that causes hardware to clear this field.
2	Reserved.
1	<b>InterceptInit: intercept INIT.</b> Read-write,Volatile. Reset: 0. 0=INIT delivered normally. 1=INIT translated into a SX interrupt. This bit controls how INIT is delivered in host mode. This bit is set by hardware when the SKINIT instruction is executed.
0	Reserved.

**MSRC001\_0115 [IGNNE] (Core::X86::Msr::IGNNE)**

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0115

Bits	Description
63:1	Reserved.
0	<b>IGNNE: current IGNNE state.</b> Read-write. Reset: 0. This bit controls the current state of the processor internal IGNNE signal.

**MSRC001\_0117 [Virtual Machine Host Save Physical Address] (Core::X86::Msr::VM\_HSAVE\_PA)**

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0117

Bits	Description
63:48	Reserved.
47:12	<b>VM_HSAVE_PA: physical address of host save area.</b> Read-write. Reset: 0_0000_0000h. This register contains the physical address of a 4-KB region where VMRUN saves host state and where vm-exit restores host state from. Writing this register causes a #GP if (FFFF_FFFF_Fh >= VM_HSAVE_PA >= FFFD_0000_0h) or if either the TSEG or ASEG regions overlap with the range defined by this register.
11:0	Reserved.

**MSRC001\_0118 [SVM Lock Key] (Core::X86::Msr::SvmLockKey)**

Read-write. Reset: Fixed,0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0118

Bits	Description
63:0	<b>SvmLockKey: SVM lock key.</b> Read-write. Reset: Fixed,0000_0000_0000_0000h. Writes to this register when (Core::X86::Msr::VM_CR[Lock] == 0) modify SvmLockKey. If ((Core::X86::Msr::VM_CR[Lock] == 1) && (SvmLockKey!=0) && (The write value == The value stored in SvmLockKey)) for a write to this register, then hardware updates Core::X86::Msr::VM_CR[Lock] = 0. Reads of this register always return zero.

**MSRC001\_011B [AVIC Doorbell] (Core::X86::Msr::AvicDoorbell)**

Write-only,Error-on-read. Reset: 0000\_0000\_0000\_0000h.

The ApicId is a physical APIC Id; not valid for logical APIC ID.

Enable: (Core::X86::Cpuid::SvmRevFeatIdEdx[AVIC] == 1).

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_011B

Bits	Description
63:32	Reserved.
31:0	<b>ApicId: APIC ID [31:0].</b> Write-only,Error-on-read. Reset: 0000_0000h. The value written must be a valid physical APID_ID.

**MSRC001\_0135 [Virtual TOM] (Core::X86::Msr::VIRTUAL\_TOM)**

Configurable. Reset: 0000\_0000\_0000\_0000h.

Access of Core::X86::Msr::VIRTUAL\_TOM in hypervisor mode causes #GP.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0135

Bits	Description
63:52	Reserved.
51:21	<b>VIRTUAL_TOM.</b> Configurable. Reset: 0000_0000h. Guest physical addresses below VIRTUAL_TOM are considered private (C=1) when VIRTUAL_TOM is enabled. Access is AccessType: (SEV_FEATURES[VirtualTom] AND SEV_FEATURES[SNPActive]) ? Read-write : Error-on-read, Error-on-write.
20:0	Reserved.

**MSRC001\_0138 [Secure AVIC Control] (Core::X86::Msr::SecureAVIC)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0138

Bits	Description
63:12	<b>GuestApicBackingPagePtr.</b> Read-write. Reset: 0_0000_0000_0000h. Guest APIC Backing Page Pointer, 4K aligned GPA address.
11:2	Reserved.
1	<b>AllowedNmi.</b> Read-write. Reset: 0. Guest allows host to send NMI.
0	<b>SecureAvicEn.</b> Read-write. Reset: 0. Secure AVIC Enable.

**MSRC001\_0140 [OS Visible Work-around Length] (Core::X86::Msr::OSVW\_ID\_Length)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0140

Bits	Description
63:16	Reserved.
15:0	<b>OSVWIdLength: OS visible work-around ID length.</b> Read-write. Reset: 0000h. See the Revision Guide for the definition of this field; see 1.2 [Reference Documents].

**MSRC001\_0141 [OS Visible Work-around Status] (Core::X86::Msr::OSVW\_Status)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0141

Bits	Description
63:0	<b>OsvwStatusBits: OS visible work-around status bits.</b> Read-write. Reset: 0000_0000_0000_0000h. See the Revision Guide for the definition of this field; see 1.2 [Reference Documents].

**MSRC001\_0200 [Performance Event Select 0] (Core::X86::Msr::PERF\_CTL0)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

See 2.1.14 [Performance Monitor Counters]. Core::X86::Msr::PERF\_LEGACY\_CTL0 is an alias of this register.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0200

Bits	Description
63:42	Reserved.
41:40	<b>HostGuestOnly: count only host/guest events.</b> Read-write. Reset: 0h. Host/Guest event counter.
<b>ValidValues:</b>	
<b>Value</b>	<b>Description</b>
0h	Count all events, irrespective of guest/host.
1h	Count guest events if [SVME] == 1.
2h	Count host events if [SVME] == 1.
3h	Count all guest and host events if [SVME] == 1.
39:36	Reserved.
35:32	<b>EventSelect[11:8].</b> Read-write. Reset: 0h. Performance event select[11:8].
31:24	<b>CntMask: counter mask.</b> Read-write. Reset: 00h. Controls the number of events counted per clock cycle.
<b>ValidValues:</b>	
<b>Value</b>	<b>Description</b>
00h	The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.
7Fh-01h	When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.
FFh-80h	Reserved.
23	<b>Inv: invert counter mask.</b> Read-write. Reset: 0. See CntMask.
22	<b>En: enable performance counter.</b> Read-write. Reset: 0. 1=Performance event counter is enabled. Performance counter enable.
21	Reserved.
20	<b>Int: enable APIC interrupt.</b> Read-write. Reset: 0. 1=APIC performance counter LVT interrupt is enabled to generate an interrupt via Core::X86::Apic::PerformanceCounterLvtEntry when the performance counter overflows. APIC interrupt enable.
19	Reserved.
18	<b>Edge: edge detect.</b> Read-write. Reset: 0. 0=Level detect. 1=Zero-to-one Edge detect. The edge count mode increments the counter when a transition happens on the monitored event. If the event selected is changed without disabling the counter, an extra edge is falsely detected when the first event is a static 0 and the second event is a static one. To avoid this false edge detection, disable the counter when changing the event and then enable the counter with a second MSR write.
17:16	<b>OsUserMode: OS and user mode.</b> Read-write. Reset: 0h. OS and user mode counter events.
<b>ValidValues:</b>	
<b>Value</b>	<b>Description</b>
0h	Count no events.
1h	Count user events (CPL>0).
2h	Count OS events (CPL=0).
3h	Count all events, irrespective of the CPL.

15:8	<b>UnitMask: event qualification.</b> Read-write. Reset: 00h. Each UnitMask bit further specifies or qualifies the event specified by EventSelect. All events selected by UnitMask are simultaneously monitored. Unless otherwise stated, the UnitMask values shown may be combined (logically ORed) to select any desired combination of the sub-events for a given event. In some cases, certain combinations can result in misleading counts, or the UnitMask value is an ordinal rather than a bit mask. These situations are described where applicable, or should be obvious from the event descriptions. For events where no UnitMask table is shown, the UnitMask is Unused.
7:0	<b>EventSelect[7:0]: event select.</b> Read-write. Reset: 00h. EventSelect[11:0] = {EventSelect[11:8], EventSelect[7:0]}. EventSelect specifies the event or event duration in a processor unit to be counted by the corresponding PERF_CTR[5:0] register. The events are specified in 2.1.14.5 [Core Performance Monitor Counters]. Some events are Reserved; when a Reserved event is selected, the results are undefined.

#### MSRC001\_020[1...B] [Performance Event Counter [5:0]] (Core::X86::Msr::PERF\_CTR)

Note: When counting events that capable of counting greater than 15 events per cycle (MergeEvent) the even and the corresponding odd PERF\_CTR must be paired to appear as a single 64-bit counter. See 2.1.14.4 [Large Increment per Cycle Events].

See Core::X86::Msr::PERF\_CTL0..5. Core::X86::Msr::PERF\_LEGACY\_CTR is an alias of MSRC001\_020[7,5,3,1]. Also can be read via x86 instructions RDPMC ECX=[05:00].

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n0; MSRC001\_0201

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n1; MSRC001\_0203

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n2; MSRC001\_0205

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n3; MSRC001\_0207

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n4; MSRC001\_0209

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n5; MSRC001\_020B

Bits	Description
63:48	Reserved.
47:0	<b>CTR.</b> Read-write, Volatile. Reset: 0000_0000_0000h. Performance counter value.



**MSRC001\_0202 [Performance Event Select 1] (Core::X86::Msr::PERF\_CTL1)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

See 2.1.14 [Performance Monitor Counters]. Core::X86::Msr::PERF\_LEGACY\_CTL1 is an alias of this register.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0202

Bits	Description										
63:42	Reserved.										
41:40	<b>HostGuestOnly: count only host/guest events.</b> Read-write. Reset: 0h. Host/Guest event counter.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Count all events, irrespective of guest/host.</td></tr> <tr> <td>1h</td><td>Count guest events if [SVME] == 1.</td></tr> <tr> <td>2h</td><td>Count host events if [SVME] == 1.</td></tr> <tr> <td>3h</td><td>Count all guest and host events if [SVME] == 1.</td></tr> </table>	Value	Description	0h	Count all events, irrespective of guest/host.	1h	Count guest events if [SVME] == 1.	2h	Count host events if [SVME] == 1.	3h	Count all guest and host events if [SVME] == 1.
Value	Description										
0h	Count all events, irrespective of guest/host.										
1h	Count guest events if [SVME] == 1.										
2h	Count host events if [SVME] == 1.										
3h	Count all guest and host events if [SVME] == 1.										
39:36	Reserved.										
35:32	<b>EventSelect[11:8].</b> Read-write. Reset: 0h. Performance event select[11:8].										
31:24	<b>CntMask: counter mask.</b> Read-write. Reset: 00h. Controls the number of events counted per clock cycle.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00h</td><td>The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.</td></tr> <tr> <td>7Fh-01h</td><td>When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.</td></tr> <tr> <td>FFh-80h</td><td>Reserved.</td></tr> </table>	Value	Description	00h	The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.	7Fh-01h	When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.	FFh-80h	Reserved.		
Value	Description										
00h	The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.										
7Fh-01h	When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.										
FFh-80h	Reserved.										
23	<b>Inv: invert counter mask.</b> Read-write. Reset: 0. See CntMask.										
22	<b>En: enable performance counter.</b> Read-write. Reset: 0. 1=Performance event counter is enabled. Performance counter enable.										
21	Reserved.										
20	<b>Int: enable APIC interrupt.</b> Read-write. Reset: 0. 1=APIC performance counter LVT interrupt is enabled to generate an interrupt via Core::X86::Apic::PerformanceCounterLvtEntry when the performance counter overflows. APIC interrupt enable.										
19	Reserved.										
18	<b>Edge: edge detect.</b> Read-write. Reset: 0. 0=Level detect. 1=Zero-to-one Edge detect. The edge count mode increments the counter when a transition happens on the monitored event. If the event selected is changed without disabling the counter, an extra edge is falsely detected when the first event is a static 0 and the second event is a static one. To avoid this false edge detection, disable the counter when changing the event and then enable the counter with a second MSR write.										
17:16	<b>OsUserMode: OS and user mode.</b> Read-write. Reset: 0h. OS and user mode counter events.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Count no events.</td></tr> <tr> <td>1h</td><td>Count user events (CPL&gt;0).</td></tr> <tr> <td>2h</td><td>Count OS events (CPL=0).</td></tr> <tr> <td>3h</td><td>Count all events, irrespective of the CPL.</td></tr> </table>	Value	Description	0h	Count no events.	1h	Count user events (CPL>0).	2h	Count OS events (CPL=0).	3h	Count all events, irrespective of the CPL.
Value	Description										
0h	Count no events.										
1h	Count user events (CPL>0).										
2h	Count OS events (CPL=0).										
3h	Count all events, irrespective of the CPL.										

15:8	<b>UnitMask: event qualification.</b> Read-write. Reset: 00h. Each UnitMask bit further specifies or qualifies the event specified by EventSelect. All events selected by UnitMask are simultaneously monitored. Unless otherwise stated, the UnitMask values shown may be combined (logically ORed) to select any desired combination of the sub-events for a given event. In some cases, certain combinations can result in misleading counts, or the UnitMask value is an ordinal rather than a bit mask. These situations are described where applicable, or should be obvious from the event descriptions. For events where no UnitMask table is shown, the UnitMask is Unused.
7:0	<b>EventSelect[7:0]: event select.</b> Read-write. Reset: 00h. EventSelect[11:0] = {EventSelect[11:8], EventSelect[7:0]}. EventSelect specifies the event or event duration in a processor unit to be counted by the corresponding PERF_CTR[5:0] register. The events are specified in 2.1.14.5 [Core Performance Monitor Counters]. Some events are Reserved; when a Reserved event is selected, the results are undefined.

**MSRC001\_0204 [Performance Event Select 2] (Core::X86::Msr::PERF\_CTL2)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

See 2.1.14 [Performance Monitor Counters]. Core::X86::Msr::PERF\_LEGACY\_CTL2 is an alias of this register.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0204

Bits	Description
63:44	Reserved.
43	<b>PreciseRetire.</b> Read-write. Reset: 0. 0=Include events counted during post-retire speculation that were later aborted. 1=Excludes events counted during post-retire speculation that were later aborted.
42	Reserved.
41:40	<b>HostGuestOnly: count only host/guest events.</b> Read-write. Reset: 0h. Host/Guest event counter.
<b>ValidValues:</b>	
Value	Description
0h	Count all events, irrespective of guest/host.
1h	Count guest events if [SVME] == 1.
2h	Count host events if [SVME] == 1.
3h	Count all guest and host events if [SVME] == 1.
39:36	Reserved.
35:32	<b>EventSelect[11:8].</b> Read-write. Reset: 0h. Performance event select[11:8].
31:24	<b>CntMask: counter mask.</b> Read-write. Reset: 00h. Controls the number of events counted per clock cycle.
<b>ValidValues:</b>	
Value	Description
00h	The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.
7Fh-01h	When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.
FFh-80h	Reserved.
23	<b>Inv: invert counter mask.</b> Read-write. Reset: 0. See CntMask.
22	<b>En: enable performance counter.</b> Read-write. Reset: 0. 1=Performance event counter is enabled. Performance counter enable.
21	Reserved.
20	<b>Int: enable APIC interrupt.</b> Read-write. Reset: 0. 1=APIC performance counter LVT interrupt is enabled to generate an interrupt via Core::X86::Apic::PerformanceCounterLvtEntry when the performance counter overflows. APIC interrupt enable.
19	Reserved.
18	<b>Edge: edge detect.</b> Read-write. Reset: 0. 0=Level detect. 1=Zero-to-one Edge detect. The edge count mode increments the counter when a transition happens on the monitored event. If the event selected is changed without disabling the counter, an extra edge is falsely detected when the first event is a static 0 and the second event is a static one. To avoid this false edge detection, disable the counter when changing the event and then enable the counter with a second MSR write.
17:16	<b>OsUserMode: OS and user mode.</b> Read-write. Reset: 0h. OS and user mode counter events.
<b>ValidValues:</b>	
Value	Description
0h	Count no events.
1h	Count user events (CPL>0).
2h	Count OS events (CPL=0).
3h	Count all events, irrespective of the CPL.

15:8	<b>UnitMask: event qualification.</b> Read-write. Reset: 00h. Each UnitMask bit further specifies or qualifies the event specified by EventSelect. All events selected by UnitMask are simultaneously monitored. Unless otherwise stated, the UnitMask values shown may be combined (logically ORed) to select any desired combination of the sub-events for a given event. In some cases, certain combinations can result in misleading counts, or the UnitMask value is an ordinal rather than a bit mask. These situations are described where applicable, or should be obvious from the event descriptions. For events where no UnitMask table is shown, the UnitMask is Unused.
7:0	<b>EventSelect[7:0]: event select.</b> Read-write. Reset: 00h. EventSelect[11:0] = {EventSelect[11:8], EventSelect[7:0]}. EventSelect specifies the event or event duration in a processor unit to be counted by the corresponding PERF_CTR[5:0] register. The events are specified in 2.1.14.5 [Core Performance Monitor Counters]. Some events are Reserved; when a Reserved event is selected, the results are undefined.

**MSRC001\_0206 [Performance Event Select 3] (Core::X86::Msr::PERF\_CTL3)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

See 2.1.14 [Performance Monitor Counters]. Core::X86::Msr::PERF\_LEGACY\_CTL3 is an alias of this register.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0206

Bits	Description										
63:42	Reserved.										
41:40	<b>HostGuestOnly: count only host/guest events.</b> Read-write. Reset: 0h. Host/Guest event counter.										
	<b>ValidValues:</b>										
	<table><tr><th>Value</th><th>Description</th></tr><tr><td>0h</td><td>Count all events, irrespective of guest/host.</td></tr><tr><td>1h</td><td>Count guest events if [SVME] == 1.</td></tr><tr><td>2h</td><td>Count host events if [SVME] == 1.</td></tr><tr><td>3h</td><td>Count all guest and host events if [SVME] == 1.</td></tr></table>	Value	Description	0h	Count all events, irrespective of guest/host.	1h	Count guest events if [SVME] == 1.	2h	Count host events if [SVME] == 1.	3h	Count all guest and host events if [SVME] == 1.
	Value	Description									
	0h	Count all events, irrespective of guest/host.									
	1h	Count guest events if [SVME] == 1.									
2h	Count host events if [SVME] == 1.										
3h	Count all guest and host events if [SVME] == 1.										
39:36	Reserved.										
35:32	<b>EventSelect[11:8].</b> Read-write. Reset: 0h. Performance event select[11:8].										
31:24	<b>CntMask: counter mask.</b> Read-write. Reset: 00h. Controls the number of events counted per clock cycle.										
	<b>ValidValues:</b>										
	<table><tr><th>Value</th><th>Description</th></tr><tr><td>00h</td><td>The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.</td></tr><tr><td>7Fh-01h</td><td>When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.</td></tr><tr><td>FFh-80h</td><td>Reserved.</td></tr></table>	Value	Description	00h	The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.	7Fh-01h	When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.	FFh-80h	Reserved.		
	Value	Description									
	00h	The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.									
7Fh-01h	When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.										
FFh-80h	Reserved.										
23	<b>Inv: invert counter mask.</b> Read-write. Reset: 0. See CntMask.										
22	<b>En: enable performance counter.</b> Read-write. Reset: 0. 1=Performance event counter is enabled. Performance counter enable.										
21	Reserved.										
20	<b>Int: enable APIC interrupt.</b> Read-write. Reset: 0. 1=APIC performance counter LVT interrupt is enabled to generate an interrupt via Core::X86::Apic::PerformanceCounterLvtEntry when the performance counter overflows. APIC interrupt enable.										
19	Reserved.										
18	<b>Edge: edge detect.</b> Read-write. Reset: 0. 0=Level detect. 1=Zero-to-one Edge detect. The edge count mode increments the counter when a transition happens on the monitored event. If the event selected is changed without disabling the counter, an extra edge is falsely detected when the first event is a static 0 and the second event is a static one. To avoid this false edge detection, disable the counter when changing the event and then enable the counter with a second MSR write.										
17:16	<b>OsUserMode: OS and user mode.</b> Read-write. Reset: 0h. OS and user mode counter events.										
	<b>ValidValues:</b>										
	<table><tr><th>Value</th><th>Description</th></tr><tr><td>0h</td><td>Count no events.</td></tr><tr><td>1h</td><td>Count user events (CPL&gt;0).</td></tr><tr><td>2h</td><td>Count OS events (CPL=0).</td></tr><tr><td>3h</td><td>Count all events, irrespective of the CPL.</td></tr></table>	Value	Description	0h	Count no events.	1h	Count user events (CPL>0).	2h	Count OS events (CPL=0).	3h	Count all events, irrespective of the CPL.
	Value	Description									
	0h	Count no events.									
	1h	Count user events (CPL>0).									
2h	Count OS events (CPL=0).										
3h	Count all events, irrespective of the CPL.										

15:8	<b>UnitMask: event qualification.</b> Read-write. Reset: 00h. Each UnitMask bit further specifies or qualifies the event specified by EventSelect. All events selected by UnitMask are simultaneously monitored. Unless otherwise stated, the UnitMask values shown may be combined (logically ORed) to select any desired combination of the sub-events for a given event. In some cases, certain combinations can result in misleading counts, or the UnitMask value is an ordinal rather than a bit mask. These situations are described where applicable, or should be obvious from the event descriptions. For events where no UnitMask table is shown, the UnitMask is Unused.
7:0	<b>EventSelect[7:0]: event select.</b> Read-write. Reset: 00h. EventSelect[11:0] = {EventSelect[11:8], EventSelect[7:0]}. EventSelect specifies the event or event duration in a processor unit to be counted by the corresponding PERF_CTR[5:0] register. The events are specified in 2.1.14.5 [Core Performance Monitor Counters]. Some events are Reserved; when a Reserved event is selected, the results are undefined.

**MSRC001\_0208 [Performance Event Select 4] (Core::X86::Msr::PERF\_CTL4)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

See 2.1.14 [Performance Monitor Counters].

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_0208

Bits	Description										
63:42	Reserved.										
41:40	<b>HostGuestOnly: count only host/guest events.</b> Read-write. Reset: 0h. Host/Guest event counters.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Count all events, irrespective of guest/host.</td></tr> <tr> <td>1h</td><td>Count guest events if [SVME] == 1.</td></tr> <tr> <td>2h</td><td>Count host events if [SVME] == 1.</td></tr> <tr> <td>3h</td><td>Count all guest and host events if [SVME] == 1.</td></tr> </table>	Value	Description	0h	Count all events, irrespective of guest/host.	1h	Count guest events if [SVME] == 1.	2h	Count host events if [SVME] == 1.	3h	Count all guest and host events if [SVME] == 1.
Value	Description										
0h	Count all events, irrespective of guest/host.										
1h	Count guest events if [SVME] == 1.										
2h	Count host events if [SVME] == 1.										
3h	Count all guest and host events if [SVME] == 1.										
39:36	Reserved.										
35:32	<b>EventSelect[11:8].</b> Read-write. Reset: 0h. Performance event select[11:8].										
31:24	<b>CntMask: counter mask.</b> Read-write. Reset: 00h. Controls the number of events counted per clock cycle.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00h</td><td>The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.</td></tr> <tr> <td>7Fh-01h</td><td>When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.</td></tr> <tr> <td>FFh-80h</td><td>Reserved.</td></tr> </table>	Value	Description	00h	The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.	7Fh-01h	When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.	FFh-80h	Reserved.		
Value	Description										
00h	The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.										
7Fh-01h	When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.										
FFh-80h	Reserved.										
23	<b>Inv: invert counter mask.</b> Read-write. Reset: 0. See CntMask.										
22	<b>En: enable performance counter.</b> Read-write. Reset: 0. 1=Performance event counter is enabled. Performance counter enable.										
21	Reserved.										
20	<b>Int: enable APIC interrupt.</b> Read-write. Reset: 0. 1=APIC performance counter LVT interrupt is enabled to generate an interrupt via Core::X86::Apic::PerformanceCounterLvtEntry when the performance counter overflows. APIC interrupt enable.										
19	Reserved.										
18	<b>Edge: edge detect.</b> Read-write. Reset: 0. 0=Level detect. 1=Zero-to-one Edge detect. The edge count mode increments the counter when a transition happens on the monitored event. If the event selected is changed without disabling the counter, an extra edge is falsely detected when the first event is a static 0 and the second event is a static one. To avoid this false edge detection, disable the counter when changing the event and then enable the counter with a second MSR write.										
17:16	<b>OsUserMode: OS and user mode.</b> Read-write. Reset: 0h. OS and user mode counter events.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Count no events.</td></tr> <tr> <td>1h</td><td>Count user events (CPL&gt;0).</td></tr> <tr> <td>2h</td><td>Count OS events (CPL=0).</td></tr> <tr> <td>3h</td><td>Count all events, irrespective of the CPL.</td></tr> </table>	Value	Description	0h	Count no events.	1h	Count user events (CPL>0).	2h	Count OS events (CPL=0).	3h	Count all events, irrespective of the CPL.
Value	Description										
0h	Count no events.										
1h	Count user events (CPL>0).										
2h	Count OS events (CPL=0).										
3h	Count all events, irrespective of the CPL.										

15:8	<b>UnitMask: event qualification.</b> Read-write. Reset: 00h. Each UnitMask bit further specifies or qualifies the event specified by EventSelect. All events selected by UnitMask are simultaneously monitored. Unless otherwise stated, the UnitMask values shown may be combined (logically ORed) to select any desired combination of the sub-events for a given event. In some cases, certain combinations can result in misleading counts, or the UnitMask value is an ordinal rather than a bit mask. These situations are described where applicable, or should be obvious from the event descriptions. For events where no UnitMask table is shown, the UnitMask is Unused.
7:0	<b>EventSelect[7:0]: event select.</b> Read-write. Reset: 00h. EventSelect[11:0] = {EventSelect[11:8], EventSelect[7:0]}. EventSelect specifies the event or event duration in a processor unit to be counted by the corresponding PERF_CTR[5:0] register. The events are specified in 2.1.14.5 [Core Performance Monitor Counters]. Some events are Reserved; when a Reserved event is selected, the results are undefined.



**MSRC001\_020A [Performance Event Select 5] (Core::X86::Msr::PERF\_CTL5)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

See 2.1.14 [Performance Monitor Counters].

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_020A

Bits	Description										
63:42	Reserved.										
41:40	<b>HostGuestOnly: count only host/guest events.</b> Read-write. Reset: 0h. Host/Guest event counter.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Count all events, irrespective of guest/host.</td></tr> <tr> <td>1h</td><td>Count guest events if [SVME] == 1.</td></tr> <tr> <td>2h</td><td>Count host events if [SVME] == 1.</td></tr> <tr> <td>3h</td><td>Count all guest and host events if [SVME] == 1.</td></tr> </table>	Value	Description	0h	Count all events, irrespective of guest/host.	1h	Count guest events if [SVME] == 1.	2h	Count host events if [SVME] == 1.	3h	Count all guest and host events if [SVME] == 1.
Value	Description										
0h	Count all events, irrespective of guest/host.										
1h	Count guest events if [SVME] == 1.										
2h	Count host events if [SVME] == 1.										
3h	Count all guest and host events if [SVME] == 1.										
39:36	Reserved.										
35:32	<b>EventSelect[11:8].</b> Read-write. Reset: 0h. Performance event select[11:8].										
31:24	<b>CntMask: counter mask.</b> Read-write. Reset: 00h. Controls the number of events counted per clock cycle.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00h</td><td>The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.</td></tr> <tr> <td>7Fh-01h</td><td>When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.</td></tr> <tr> <td>FFh-80h</td><td>Reserved.</td></tr> </table>	Value	Description	00h	The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.	7Fh-01h	When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.	FFh-80h	Reserved.		
Value	Description										
00h	The corresponding PERF_CTR[5:0] register increments by the number of events occurring in a clock cycle. See 2.1.14.4 [Large Increment per Cycle Events] for events that can increment greater than 15 per cycle.										
7Fh-01h	When Inv == 0, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is greater than or equal to the CntMask value. When Inv == 1, the corresponding PERF_CTR[5:0] register increments by 1, if the number of events occurring in a clock cycle is less than CntMask value.										
FFh-80h	Reserved.										
23	<b>Inv: invert counter mask.</b> Read-write. Reset: 0. See CntMask.										
22	<b>En: enable performance counter.</b> Read-write. Reset: 0. 1=Performance event counter is enabled. Performance counter enable.										
21	Reserved.										
20	<b>Int: enable APIC interrupt.</b> Read-write. Reset: 0. 1=APIC performance counter LVT interrupt is enabled to generate an interrupt via Core::X86::Apic::PerformanceCounterLvtEntry when the performance counter overflows. APIC interrupt enable.										
19	Reserved.										
18	<b>Edge: edge detect.</b> Read-write. Reset: 0. 0=Level detect. 1=Zero-to-one Edge detect. The edge count mode increments the counter when a transition happens on the monitored event. If the event selected is changed without disabling the counter, an extra edge is falsely detected when the first event is a static 0 and the second event is a static one. To avoid this false edge detection, disable the counter when changing the event and then enable the counter with a second MSR write.										
17:16	<b>OsUserMode: OS and user mode.</b> Read-write. Reset: 0h. OS and user mode counter events.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Count no events.</td></tr> <tr> <td>1h</td><td>Count user events (CPL&gt;0).</td></tr> <tr> <td>2h</td><td>Count OS events (CPL=0).</td></tr> <tr> <td>3h</td><td>Count all events, irrespective of the CPL.</td></tr> </table>	Value	Description	0h	Count no events.	1h	Count user events (CPL>0).	2h	Count OS events (CPL=0).	3h	Count all events, irrespective of the CPL.
Value	Description										
0h	Count no events.										
1h	Count user events (CPL>0).										
2h	Count OS events (CPL=0).										
3h	Count all events, irrespective of the CPL.										

15:8	<b>UnitMask: event qualification.</b> Read-write. Reset: 00h. Each UnitMask bit further specifies or qualifies the event specified by EventSelect. All events selected by UnitMask are simultaneously monitored. Unless otherwise stated, the UnitMask values shown may be combined (logically ORed) to select any desired combination of the sub-events for a given event. In some cases, certain combinations can result in misleading counts, or the UnitMask value is an ordinal rather than a bit mask. These situations are described where applicable, or should be obvious from the event descriptions. For events where no UnitMask table is shown, the UnitMask is Unused.
7:0	<b>EventSelect[7:0]: event select.</b> Read-write. Reset: 00h. EventSelect[11:0] = {EventSelect[11:8], EventSelect[7:0]}. EventSelect specifies the event or event duration in a processor unit to be counted by the corresponding PERF_CTR[5:0] register. The events are specified in 2.1.14.5 [Core Performance Monitor Counters]. Some events are Reserved; when a Reserved event is selected, the results are undefined.

**MSRC001\_023[0...A] [L3 Performance Event Select [5:0]] (Core::X86::Msr::ChL3PmcCfg)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

See 2.1.14.6 [L3 Cache Performance Monitor Counters]

\_lthree[1:0]\_n0; MSRC001\_0230

\_lthree[1:0]\_n1; MSRC001\_0232

\_lthree[1:0]\_n2; MSRC001\_0234

\_lthree[1:0]\_n3; MSRC001\_0236

\_lthree[1:0]\_n4; MSRC001\_0238

\_lthree[1:0]\_n5; MSRC001\_023A

Bits	Description								
63:60	Reserved.								
59:56	<b>ThreadMask.</b> Read-write. Reset: 0h. Controls which of the 2 threads in the selected core are being counted. In non-SMT mode, thread 0 must be selected. One or more threads must be selected unless otherwise specified by the specific L3PMC event. <b>ValidValues:</b> <table> <tr> <th>Bit</th><th>Description</th></tr> <tr> <td>[0]</td><td>Thread 0.</td></tr> <tr> <td>[1]</td><td>Thread 1.</td></tr> <tr> <td>[3:2]</td><td>Reserved.</td></tr> </table>	Bit	Description	[0]	Thread 0.	[1]	Thread 1.	[3:2]	Reserved.
Bit	Description								
[0]	Thread 0.								
[1]	Thread 1.								
[3:2]	Reserved.								
55:51	Reserved.								
50:48	<b>SourceId.</b> Read-write. Reset: 0h. For L3 PMC Events, controls the L3 slice for which events are counted. For Xi PMC Events, controls the CCX interface for which events are counted. Unless otherwise noted by the specific L3PMC event, use Core::X86::Msr::ChL3PmcCfg[SourceId] to select an individual Slice/CCX Interface or Core::X86::Msr::ChL3PmcCfg[EnAllSources] to select all Slices/CCX Interfaces. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>1h-0h</td><td>L3 PMC Event: &lt;Value&gt; Slice, Xi PMC Event: &lt;Value&gt; CCX Interface</td></tr> <tr> <td>7h-2h</td><td>L3 PMC Event: &lt;Value&gt; Slice, Xi PMC Event: Reserved</td></tr> </table>	Value	Description	1h-0h	L3 PMC Event: <Value> Slice, Xi PMC Event: <Value> CCX Interface	7h-2h	L3 PMC Event: <Value> Slice, Xi PMC Event: Reserved		
Value	Description								
1h-0h	L3 PMC Event: <Value> Slice, Xi PMC Event: <Value> CCX Interface								
7h-2h	L3 PMC Event: <Value> Slice, Xi PMC Event: Reserved								
47	<b>EnAllCores.</b> Read-write. Reset: 0. 1=Enable counting L3 events for all cores simultaneously.								
46	<b>EnAllSources.</b> Read-write. Reset: 0. 1=Enable counting. For L3 PMC Events, enable counting on L3 slices simultaneously. For Xi PMC Events, enable counting on all CCX interfaces simultaneously.								
45	Reserved.								
44:42	<b>CoreId.</b> Read-write. Reset: 0h. Controls core for which events are to be counted. See Core::X86::Msr::ChL3PmcCfg[EnAllCores] to count all cores simultaneously. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>7h-0h</td><td>&lt;Value&gt; CoreId.</td></tr> </table>	Value	Description	7h-0h	<Value> CoreId.				
Value	Description								
7h-0h	<Value> CoreId.								
41:23	Reserved.								
22	<b>Enable: Enable L3 performance counter.</b> Read-write. Reset: 0. 1=Enable.								
21:16	Reserved.								
15:8	<b>UnitMask: event qualification.</b> Read-write. Reset: 00h. Each UnitMask bit further specifies or qualifies the event specified by EventSelect. All events selected by UnitMask are simultaneously monitored. Unless otherwise stated, the UnitMask values shown may be combined (logically ORed) to select any desired combination of the sub-events for a given event. In some cases, certain combinations can result in misleading counts, or the UnitMask value is an ordinal rather than a bit mask. These situations are described where applicable, or should be obvious from the event descriptions. For events where no UnitMask table is shown, the UnitMask is Unused. When selecting an event for which not all UnitMask bits are defined, the undefined UnitMask bits should be set to zero.								
7:0	<b>EventSel.</b> Read-write. Reset: 00h. L3 Event select.								

**MSRC001\_02[40...5E] [Data Fabric Performance Event Select [3:0]] (Core::X86::Msr::DF\_PERF\_CTL)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

See 2.1.14 [Performance Monitor Counters].

The DF Performance Monitors are shared by all cores/threads in the node. See 2.1.9 [Register Sharing].

\_n0; MSRC001\_0240

\_n1; MSRC001\_0242

\_n2; MSRC001\_0244

\_n3; MSRC001\_0246

\_n4; MSRC001\_0248

\_n5; MSRC001\_024A

\_n6; MSRC001\_024C

\_n7; MSRC001\_024E

\_n8; MSRC001\_0250

\_n9; MSRC001\_0252

\_n10; MSRC001\_0254

\_n11; MSRC001\_0256

\_n12; MSRC001\_0258

\_n13; MSRC001\_025A

\_n14; MSRC001\_025C

\_n15; MSRC001\_025E

Bits	Description
63:38	Reserved.
37:32	<b>EventSelect[13:8]: performance event select.</b> Read-write. Reset: 00h. Performance event select [13:0]. See EventSelect[7:0].
31:28	Reserved.
27:24	<b>UnitMask[11:8]: event qualification.</b> Read-write. Reset: 0h. Uppper 4 bits of UnitMask. See UnitMask[7:0].
23	Reserved.
22	<b>En: enable performance counter.</b> Read-write. Reset: 0. 1=Performance event counter is enabled.
21:16	Reserved.
15:8	<b>UnitMask[7:0]: event qualification.</b> Read-write. Reset: 00h. Each UnitMask bit further specifies or qualifies the event specified by EventSelect. All events selected by UnitMask are simultaneously monitored.
7:0	<b>EventSelect[7:0]: event select.</b> Read-write. Reset: 00h. This field, along with EventSelect[13:8] above, combine to form the 14-bit event select field, EventSelect[13:0]. EventSelect specifies the event or event duration in a processor unit to be counted by the corresponding DF_PERF_CTR[3:0] register. Some events are Reserved; when a Reserved event is selected, the results are undefined.

**MSRC001\_02[41...5F] [Data Fabric Performance Event Counter [3:0]] (Core::X86::Msr::DF\_PERF\_CTR)**

See Core::X86::Msr::DF\_PERF\_CTL. Also can be read via x86 instructions RDPMS ECX=[09:06].  
The DF Performance Monitors are shared by all cores/threads in the socket.

_n0; MSRC001_0241	
_n1; MSRC001_0243	
_n2; MSRC001_0245	
_n3; MSRC001_0247	
_n4; MSRC001_0249	
_n5; MSRC001_024B	
_n6; MSRC001_024D	
_n7; MSRC001_024F	
_n8; MSRC001_0251	
_n9; MSRC001_0253	
_n10; MSRC001_0255	
_n11; MSRC001_0257	
_n12; MSRC001_0259	
_n13; MSRC001_025B	
_n14; MSRC001_025D	
_n15; MSRC001_025F	
Bits	Description
63:48	Reserved.
47:0	<b>CTR[47:0]: performance counter value[47:0].</b> Read-write,Volatile. Reset: 0000_0000_0000h. The current value of the event counter.

**MSRC001\_029A [Core Energy Status] (Core::X86::Msr::CORE\_ENERGY\_STAT)**

Read-only, Volatile. Reset: 0000\_0000\_0000\_0000h.

_lthree[1:0]_core[7:0]; MSRC001_029A	
Bits	Description
63:0	<b>TotalEnergyConsumed.</b> Read-only,Volatile. Reset: 0000_0000_0000_0000h.

**MSRC001\_02B0 [CPPC Capability 1] (Core::X86::Msr::CpccCapability1)**

Collaborative Processor Performance Control Capability 1.

_lthree[1:0]_core[7:0]_thread[1:0]; MSRC001_02B0	
Bits	Description
63:32	Reserved.
31:24	<b>HighestPerf: Highest Performance.</b> Read-only,Error-on-write,Volatile. Reset: 00h. Value for the maximum non-ensured performance level.
23:16	<b>NominalPerf: Nominal Performance.</b> Read-only,Error-on-write,Volatile. Reset: 00h. Value for the maximum sustained performance level assuming ideal operating conditions.
15:8	<b>LowNonLinPerf: Lowest Nonlinear Performance.</b> Read-only,Error-on-write,Volatile. Reset: 00h. Value for the lowest nonlinear performance level.
7:0	<b>LowestPerf: Lowest Performance.</b> Read-only,Error-on-write,Volatile. Reset: 00h. Value for the lowest performance level that software can program to MSR_CPPC_REQUEST.

**MSRC001\_02B1 [CPPC Enable] (Core::X86::Msr::CpccEnable)**

Collaborative Processor Performance Control Enable.

MSRC001_02B1	
Bits	Description
63:1	Reserved.
0	<b>CppcEnable.</b> Read,Write-1-only. Reset: 0. CPPC Enable.

**MSRC001\_02B2 [CPPC Capability 2] (Core::X86::Msr::CpccCapability2)**

Collaborative Processor Performance Control Capability 2.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_02B2

Bits	Description
63:8	Reserved.
7:0	<b>MaxPerf: constrained maximum performance.</b> Read-only, Error-on-write, Volatile. Reset: 00h. Value for the current maximum performance level taking into account all known external constraints (i.e., power limits, thermal limits, AC/DC power source, etc.).

**MSRC001\_02B3 [CPPC Request] (Core::X86::Msr::CpccRequest)**

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_02B3

Bits	Description
63:32	Reserved.
31:24	<b>EnergyPerfPref.</b> Read-write. Reset: 00h. Energy Performance Preference.
23:16	<b>DesPerf.</b> Read-write. Reset: 00h. Desired Performance.
15:8	<b>MinPerf.</b> Read-write. Reset: 00h. Minimum Performance.
7:0	<b>MaxPerf.</b> Read-write. Reset: 00h. Maximum Performance.

**MSRC001\_02B4 [CPPC Status] (Core::X86::Msr::CpccStatus)**

Reset: 0000\_0000\_0000\_0000h.

Collaborative Processor Performance Control Status.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_02B4

Bits	Description
63:2	Reserved.
1	<b>MINEXCURSION.</b> Read-write, Volatile. Reset: 0. 1=Minimum Excursion has occurred.
0	Reserved.

**MSRC001\_02F0 [Protected Processor Inventory Number Control] (Core::X86::Msr::PPIN\_CTL)**

Unpredictable.

MSRC001\_02F0

Bits	Description
63:2	Reserved.
1	<b>PPIN_EN.</b> Unpredictable. Reset: X. 0=Reading Core::X86::Msr::PPIN will cause a #GP. 1=Core::X86::Msr::PPIN is accessible using RDMSR. Once set, attempting to write 1 to Core::X86::Msr::PPIN_CTL[Lockout] will cause a #GP.
0	<b>Lockout.</b> Unpredictable. Reset: X. 0=Writes to Core::X86::Msr::PPIN_CTL are permitted if PPIN_EN=0. 1=Further writes to Core::X86::Msr::PPIN_CTL are ignored.  <b>Description:</b> Writing 1 to Core::X86::Msr::PPIN_CTL[Lockout] is permitted only if Core::X86::Msr::PPIN_CTL[PPIN_EN] == 0. BIOS should provide an opt-in menu to enable the user to turn on Core::X86::Msr::PPIN_CTL[PPIN_EN] for privileged inventory initialization agent to access Core::X86::Msr::PPIN. After reading Core::X86::Msr::PPIN, the privileged inventory initialization agent should write 00b followed by 01b to Core::X86::Msr::PPIN_CTL to disable further access to MSR_PPIN and prevent unauthorized modification to MSR_PPIN_CTL. Once this bit is written with 1, subsequent writes to this register are ignored, and a reset (warm or cold) is required in order to clear it, which gives BIOS the opportunity to set it again at the next boot.

**MSRC001\_02F1 [Protected Processor Inventory Number] (Core::X86::Msr::PPIN)**

MSRC001\_02F1

Bits	Description
63:0	<b>PPIN.</b> Reset: Fixed,XXXX_XXXX_XXXX_XXXXh. Protected Processor Inventory Number. AccessType: ({Core::X86::Msr::PPIN_CTL[PPIN_EN] , Core::X86::Msr::PPIN_CTL[Lockout]} == 2h) ? Read,Error-on-write : Error-on-read,Error-on-write.

**MSRC001\_03[00...1E] [Access of From IP on Last Branch Record Stack] (Core::X86::Msr::LastBranchStackFromIp)**

Reset: 0000\_0000\_0000\_0000h.

These MSRs capture the branch from IP when LBR V2 is enabled (see Core::X86::Msr::DebugExtnCtl[LBRV2EN]). The youngest branch is in the lowest numbered MSR and the oldest branch is in the highest numbered MSR.

_lthree[1:0]_core[7:0]_thread[1:0]_n0; MSRC001_0300	
_lthree[1:0]_core[7:0]_thread[1:0]_n1; MSRC001_0302	
_lthree[1:0]_core[7:0]_thread[1:0]_n2; MSRC001_0304	
_lthree[1:0]_core[7:0]_thread[1:0]_n3; MSRC001_0306	
_lthree[1:0]_core[7:0]_thread[1:0]_n4; MSRC001_0308	
_lthree[1:0]_core[7:0]_thread[1:0]_n5; MSRC001_030A	
_lthree[1:0]_core[7:0]_thread[1:0]_n6; MSRC001_030C	
_lthree[1:0]_core[7:0]_thread[1:0]_n7; MSRC001_030E	
_lthree[1:0]_core[7:0]_thread[1:0]_n8; MSRC001_0310	
_lthree[1:0]_core[7:0]_thread[1:0]_n9; MSRC001_0312	
_lthree[1:0]_core[7:0]_thread[1:0]_n10; MSRC001_0314	
_lthree[1:0]_core[7:0]_thread[1:0]_n11; MSRC001_0316	
_lthree[1:0]_core[7:0]_thread[1:0]_n12; MSRC001_0318	
_lthree[1:0]_core[7:0]_thread[1:0]_n13; MSRC001_031A	
_lthree[1:0]_core[7:0]_thread[1:0]_n14; MSRC001_031C	
_lthree[1:0]_core[7:0]_thread[1:0]_n15; MSRC001_031E	
Bits	Description
63	<b>Mispredict.</b> Read-write. Reset: 0. Indicates if the recorded branch mispredicted.
62:58	<b>BranchFromIpSignExt.</b> Read-write,Volatile. Reset: 00h. Sign extension of BranchFromIp.
57:0	<b>BranchFromIp.</b> Read-write,Volatile. Reset: 000_0000_0000_0000h. Either the lower 63b of the branch segment offset or the segment offset of an instruction preceding the branch. If the segment offset recorded is not the start of the branch instruction, it is the start of a non-branch instruction up to 16 bytes before the end of the branch instruction. The next branch in the sequential path after this instruction is the branch for this LBR Stack entry. Not recording the segment offset of the branch is the result of instruction fusion. If it is desired that BranchFromIp always records the address of branches Instruction Fusion needs to be disabled by also enabling Legacy LBR via Core::X86::Msr::DBG_CTL_MSR[LBR].

**MSRC001\_03[01...1F] [Access of To IP on Last Branch Record Stack] (Core::X86::Msr::LastBranchStackToIp)**

Reset: 0000\_0000\_0000\_0000h.

These MSRs capture the branch to IP and additional information when LBR V2 is enabled (see Core::X86::Msr::DebugExtnCtl[LBRV2EN]). The youngest branch is in the lowest numbered MSR and the oldest branch is in the highest numbered MSR.

The following table shows the types of branch records based on the Valid and Spec bits:

Valid	Spec	Description
1	0	Normal recorded branch.
1	1	Branch was recorded when speculative performance feature was active.
0	1	Branch was recorded but speculative feature was not successful.
0	0	No branch has yet been logged in this entry since software last cleared this MSR by writing 0.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n0; MSRC001\_0301

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n1; MSRC001\_0303

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n2; MSRC001\_0305

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n3; MSRC001\_0307

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n4; MSRC001\_0309

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n5; MSRC001\_030B

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n6; MSRC001\_030D

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n7; MSRC001\_030F

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n8; MSRC001\_0311

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n9; MSRC001\_0313

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n10; MSRC001\_0315

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n11; MSRC001\_0317

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n12; MSRC001\_0319

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n13; MSRC001\_031B

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n14; MSRC001\_031D

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_n15; MSRC001\_031F

Bits	Description
63	<b>Valid.</b> Read-write. Reset: 0. The valid bit is cleared when the branch was speculatively recorded by the hardware but eventually discarded. Valid LastBranchFrom/To entries have this bit set.
62	<b>Spec: Speculative.</b> Read-write. Reset: 0. When set, the entry was written speculatively .
61	Reserved.
60:58	<b>BranchToIpSignExt.</b> Read-write, Volatile. Reset: 0h. Sign extension of BranchToIp.
57:0	<b>BranchToIp.</b> Read-write, Volatile. Reset: 000_0000_0000_0000h. Lower 58b of the branch target code segment offset.



**MSRC001\_08[00...7E] [UMC Performance Monitor Control] (Core::X86::MsR::UMC\_PerfMonCtl)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

These MSRs provide access to the UMC Performance Monitor Control registers. Refer to Core::X86::Cpuid::ExtPerfMonAndDbgEbx[NumPerfCtrUmc] and Core::X86::Cpuid::ExtPerfMonAndDbgEcx[ActiveUmcMask] for the number of available UMC PMCs and the number of PMCs per UMC. MSRs belonging to non-existent UMC Performance Monitors return zero when read and ignore writes. Reserved bits must be set to 0 for reliable results.

\_n0; MSRC001\_0800

\_n1; MSRC001\_0802

\_n2; MSRC001\_0804

\_n3; MSRC001\_0806

\_n4; MSRC001\_0808

\_n5; MSRC001\_080A

\_n6; MSRC001\_080C

\_n7; MSRC001\_080E

\_n8; MSRC001\_0810

\_n9; MSRC001\_0812

\_n10; MSRC001\_0814

\_n11; MSRC001\_0816

\_n12; MSRC001\_0818

\_n13; MSRC001\_081A

\_n14; MSRC001\_081C

\_n15; MSRC001\_081E

\_n16; MSRC001\_0820

\_n17; MSRC001\_0822

\_n18; MSRC001\_0824

\_n19; MSRC001\_0826

\_n20; MSRC001\_0828

\_n21; MSRC001\_082A

\_n22; MSRC001\_082C

\_n23; MSRC001\_082E

\_n24; MSRC001\_0830

\_n25; MSRC001\_0832

\_n26; MSRC001\_0834

\_n27; MSRC001\_0836

\_n28; MSRC001\_0838

\_n29; MSRC001\_083A

\_n30; MSRC001\_083C

\_n31; MSRC001\_083E

\_n32; MSRC001\_0840

\_n33; MSRC001\_0842

\_n34; MSRC001\_0844

\_n35; MSRC001\_0846

\_n36; MSRC001\_0848

\_n37; MSRC001\_084A

\_n38; MSRC001\_084C

\_n39; MSRC001\_084E

\_n40; MSRC001\_0850

\_n41; MSRC001\_0852

\_n42; MSRC001\_0854

\_n43; MSRC001\_0856

\_n44; MSRC001\_0858

\_n45; MSRC001\_085A

\_n46; MSRC001\_085C

\_n47; MSRC001\_085E

\_n48; MSRC001\_0860

\_n49; MSRC001\_0862

\_n50; MSRC001\_0864

\_n51; MSRC001\_0866

\_n52; MSRC001\_0868

\_n53; MSRC001\_086A

\_n54; MSRC001\_086C

\_n55; MSRC001\_086E

_n56; MSRC001_0870											
_n57; MSRC001_0872											
_n58; MSRC001_0874											
_n59; MSRC001_0876											
_n60; MSRC001_0878											
_n61; MSRC001_087A											
_n62; MSRC001_087C											
_n63; MSRC001_087E											
Bits	Description										
63:32	Reserved.										
31	<b>Enable.</b> Read-write. Reset: 0. 0=Disable. 1=Enable. Counter enable.										
30:10	Reserved.										
9:8	<b>RdWrMask.</b> Read-write. Reset: 0h. Mask transactions based on read or write.										
	<b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>No masking, includes reads and writes.</td></tr> <tr> <td>1h</td><td>Mask writes.</td></tr> <tr> <td>2h</td><td>Mask reads.</td></tr> <tr> <td>3h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	No masking, includes reads and writes.	1h	Mask writes.	2h	Mask reads.	3h	Reserved.
Value	Description										
0h	No masking, includes reads and writes.										
1h	Mask writes.										
2h	Mask reads.										
3h	Reserved.										
7:0	<b>EventSelect.</b> Read-write. Reset: 00h. Select the performance monitor event.										

**MSRC001\_08[01...7F] [UMC Performance Monitor Counter] (Core::X86::Msr::UMC\_PerfMonCntr)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

These MSRs provide access to the UMC Performance Monitor Counter registers. Refer to Core::X86::Cpuid::ExtPerfMonAndDbgEbx[NumPerfCtrUmc] and Core::X86::Cpuid::ExtPerfMonAndDbgEcx[ActiveUmcMask] for the number of available UMC PMCs and the number of PMCs per UMC. MSRs belonging to non-existent UMC Performance Monitors return zero when read and ignore writes.

\_n0; MSRC001\_0801

\_n1; MSRC001\_0803

\_n2; MSRC001\_0805

\_n3; MSRC001\_0807

\_n4; MSRC001\_0809

\_n5; MSRC001\_080B

\_n6; MSRC001\_080D

\_n7; MSRC001\_080F

\_n8; MSRC001\_0811

\_n9; MSRC001\_0813

\_n10; MSRC001\_0815

\_n11; MSRC001\_0817

\_n12; MSRC001\_0819

\_n13; MSRC001\_081B

\_n14; MSRC001\_081D

\_n15; MSRC001\_081F

\_n16; MSRC001\_0821

\_n17; MSRC001\_0823

\_n18; MSRC001\_0825

\_n19; MSRC001\_0827

\_n20; MSRC001\_0829

\_n21; MSRC001\_082B

\_n22; MSRC001\_082D

\_n23; MSRC001\_082F

\_n24; MSRC001\_0831

\_n25; MSRC001\_0833

\_n26; MSRC001\_0835

\_n27; MSRC001\_0837

\_n28; MSRC001\_0839

\_n29; MSRC001\_083B

\_n30; MSRC001\_083D

\_n31; MSRC001\_083F

\_n32; MSRC001\_0841

\_n33; MSRC001\_0843

\_n34; MSRC001\_0845

\_n35; MSRC001\_0847

\_n36; MSRC001\_0849

\_n37; MSRC001\_084B

\_n38; MSRC001\_084D

\_n39; MSRC001\_084F

\_n40; MSRC001\_0851

\_n41; MSRC001\_0853

\_n42; MSRC001\_0855

\_n43; MSRC001\_0857

\_n44; MSRC001\_0859

\_n45; MSRC001\_085B

\_n46; MSRC001\_085D

\_n47; MSRC001\_085F

\_n48; MSRC001\_0861

\_n49; MSRC001\_0863

\_n50; MSRC001\_0865

\_n51; MSRC001\_0867

\_n52; MSRC001\_0869

\_n53; MSRC001\_086B

\_n54; MSRC001\_086D

\_n55; MSRC001\_086F

_n56; MSRC001_0871	
_n57; MSRC001_0873	
_n58; MSRC001_0875	
_n59; MSRC001_0877	
_n60; MSRC001_0879	
_n61; MSRC001_087B	
_n62; MSRC001_087D	
_n63; MSRC001_087F	
Bits	Description
63:49	Reserved.
48	<b>Overflow.</b> Read-write. Reset: 0. Performance Counter Overflow bit. Write-0-to-clear.
47:0	<b>Data.</b> Read-write. Reset: 0000_0000_0000h. Performance Counter Value.

#### 2.1.13.4 MSRs - MSRC001\_1xxx

##### MSRC001\_1003 [Thermal and Power Management CPUID Features] (Core::X86::Msr::CPUID\_PWR\_THERM)

Read-write.

Core::X86::Msr::CPUID\_PWR\_THERM provides control over values read from

Core::X86::CpuId::ThermalPwrMgmtEcX.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_1003

Bits	Description
63:1	Reserved.
0	<b>EffFreq.</b> Read-write. Reset: Core::X86::CpuId::ThermalPwrMgmtEcX[EffFreq]. See Core::X86::CpuId::ThermalPwrMgmtEcX[EffFreq].

**MSRC001\_1004 [CPUID Features for CPUID Fn00000001\_E[C,D]X] (Core::X86::Msr::CPUID\_Features)**

Read-write.

Core::X86::Msr::CPUID\_Features[63:32] provides control over values read from Core::X86::Cpuid::FeatureIdEcX;  
 Core::X86::Msr::CPUID\_Features[31:0] provides control over values read from Core::X86::Cpuid::FeatureIdEdX.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_1004

Bits	Description
63	Reserved.
62	<b>RDRAND</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEcX[RDRAND]. See Core::X86::Cpuid::FeatureIdEcX[RDRAND].
61	<b>F16C</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEcX[F16C]. See Core::X86::Cpuid::FeatureIdEcX[F16C].
60	<b>AVX</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEcX[AVX]. See Core::X86::Cpuid::FeatureIdEcX[AVX].
59	<b>OSXSAVE</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEcX[OSXSAVE]. Modifies Core::X86::Cpuid::FeatureIdEcX[OSXSAVE] only if CR4[OSXSAVE].
58	<b>XSAVE</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEcX[XSAVE]. See Core::X86::Cpuid::FeatureIdEcX[XSAVE].
57	<b>AES</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEcX[AES]. Modifies Core::X86::Cpuid::FeatureIdEcX[AES] only if the reset value is 1 .
56	Reserved.
55	<b>POPCNT</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEcX[POPCNT]. See Core::X86::Cpuid::FeatureIdEcX[POPCNT].
54	<b>MOVBE</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEcX[MOVBE]. See Core::X86::Cpuid::FeatureIdEcX[MOVBE].
53	<b>X2APIC</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEcX[X2APIC]. See Core::X86::Cpuid::FeatureIdEcX[X2APIC].
52	<b>SSE42</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEcX[SSE42]. See Core::X86::Cpuid::FeatureIdEcX[SSE42].
51	<b>SSE41</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEcX[SSE41]. See Core::X86::Cpuid::FeatureIdEcX[SSE41].
50:46	Reserved.
45	<b>CMPXCHG16B</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEcX[CMXCHG16B]. See Core::X86::Cpuid::FeatureIdEcX[CMXCHG16B].
44	<b>FMA</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEcX[FMA]. See Core::X86::Cpuid::FeatureIdEcX[FMA].
43:42	Reserved.
41	<b>SSSE3</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEcX[SSSE3]. See Core::X86::Cpuid::FeatureIdEcX[SSSE3].
40:36	Reserved.
35	<b>Monitor</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEcX[Monitor]. Modifies Core::X86::Cpuid::FeatureIdEcX[Monitor] only if ~Core::X86::Msr::HWCR[MonMwaitDis].
34	Reserved.
33	<b>PCLMULQDQ</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEcX[PCLMULQDQ]. Modifies Core::X86::Cpuid::FeatureIdEcX[PCLMULQDQ] only if the reset value is 1.
32	<b>SSE3</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEcX[SSE3]. See Core::X86::Cpuid::FeatureIdEcX[SSE3].
31:29	Reserved.
28	<b>HTT</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEdX[HTT]. See Core::X86::Cpuid::FeatureIdEdX[HTT].
27	Reserved.
26	<b>SSE2</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEdX[SSE2]. See Core::X86::Cpuid::FeatureIdEdX[SSE2].
25	<b>SSE</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEdX[SSE]. See Core::X86::Cpuid::FeatureIdEdX[SSE].
24	<b>FXSR</b> . Read-write. Reset: Core::X86::Cpuid::FeatureIdEdX[FXSR]. See Core::X86::Cpuid::FeatureIdEdX[FXSR].

23	<b>MMX: MMX instructions.</b> Read-write. Reset: Core::X86::Cpuid::FeatureIdEdx[MMX]. See Core::X86::Cpuid::FeatureIdEdx[MMX].
22:20	Reserved.
19	<b>CLFSH.</b> Read-write. Reset: Core::X86::Cpuid::FeatureIdEdx[CLFSH]. See Core::X86::Cpuid::FeatureIdEdx[CLFSH].
18	Reserved.
17	<b>PSE36.</b> Read-write. Reset: Core::X86::Cpuid::FeatureIdEdx[PSE36]. See Core::X86::Cpuid::FeatureIdEdx[PSE36].
16	<b>PAT.</b> Read-write. Reset: Core::X86::Cpuid::FeatureIdEdx[PAT]. See Core::X86::Cpuid::FeatureIdEdx[PAT].
15	<b>CMOV.</b> Read-write. Reset: Core::X86::Cpuid::FeatureIdEdx[CMOV]. See Core::X86::Cpuid::FeatureIdEdx[CMOV].
14	<b>MCA.</b> Read-write. Reset: Core::X86::Cpuid::FeatureIdEdx[MCA]. See Core::X86::Cpuid::FeatureIdEdx[MCA].
13	<b>PGE.</b> Read-write. Reset: Core::X86::Cpuid::FeatureIdEdx[PGE]. See Core::X86::Cpuid::FeatureIdEdx[PGE].
12	<b>MTRR.</b> Read-write. Reset: Core::X86::Cpuid::FeatureIdEdx[MTRR]. See Core::X86::Cpuid::FeatureIdEdx[MTRR].
11	<b>SysEnterSysExit.</b> Read-write. Reset: Core::X86::Cpuid::FeatureIdEdx[SysEnterSysExit]. See Core::X86::Cpuid::FeatureIdEdx[SysEnterSysExit].
10	Reserved.
9	<b>APIC.</b> Read-write. Reset: Core::X86::Cpuid::FeatureIdEdx[APIC]. Modifies Core::X86::Cpuid::FeatureIdEdx[APIC] only if Core::X86::Msr::APIC_BAR[ApicEn].
8	<b>CMPXCHG8B.</b> Read-write. Reset: Core::X86::Cpuid::FeatureIdEdx[CMXCHG8B]. See Core::X86::Cpuid::FeatureIdEdx[CMXCHG8B].
7	<b>MCE.</b> Read-write. Reset: Core::X86::Cpuid::FeatureIdEdx[MCE]. See Core::X86::Cpuid::FeatureIdEdx[MCE].
6	<b>PAE.</b> Read-write. Reset: Core::X86::Cpuid::FeatureIdEdx[PAE]. See Core::X86::Cpuid::FeatureIdEdx[PAE].
5	<b>MSR.</b> Read-write. Reset: Core::X86::Cpuid::FeatureIdEdx[MSR]. See Core::X86::Cpuid::FeatureIdEdx[MSR].
4	<b>TSC.</b> Read-write. Reset: Core::X86::Cpuid::FeatureIdEdx[TSC]. See Core::X86::Cpuid::FeatureIdEdx[TSC].
3	<b>PSE.</b> Read-write. Reset: Core::X86::Cpuid::FeatureIdEdx[PSE]. See Core::X86::Cpuid::FeatureIdEdx[PSE].
2	<b>DE.</b> Read-write. Reset: Core::X86::Cpuid::FeatureIdEdx[DE]. See Core::X86::Cpuid::FeatureIdEdx[DE].
1	<b>VME.</b> Read-write. Reset: Core::X86::Cpuid::FeatureIdEdx[VME]. See Core::X86::Cpuid::FeatureIdEdx[VME].
0	<b>FPU.</b> Read-write. Reset: Core::X86::Cpuid::FeatureIdEdx[FPU]. See Core::X86::Cpuid::FeatureIdEdx[FPU].

**MSRC001\_1005 [CPUID Features for CPUID Fn80000001\_E[C,D]X] (Core::X86::Msr::CPUID\_ExtFeatures)**

Read-write.

Core::X86::Msr::CPUID\_ExtFeatures[63:32] provides control over values read from

Core::X86::Cpuid::FeatureExtIdEcX; Core::X86::Msr::CPUID\_ExtFeatures[31:0] provides control over values read from Core::X86::Cpuid::FeatureExtIdEdX.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_1005

Bits	Description
63	Reserved.
62	<b>AdMskExtN</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcX[AdMskExtN]. See Core::X86::Cpuid::FeatureExtIdEcX[AdMskExtN].
61	<b>MwaitExtended</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcX[MwaitExtended]. See Core::X86::Cpuid::FeatureExtIdEcX[MwaitExtended].
60	<b>PerfCtrExtLLC</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcX[PerfCtrExtLLC]. See Core::X86::Cpuid::FeatureExtIdEcX[PerfCtrExtLLC].
59	<b>PerfTsc</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcX[PerfTsc]. See Core::X86::Cpuid::FeatureExtIdEcX[PerfTsc].
58	<b>DataBreakpointExtension</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcX[DataBreakpointExtension]. See Core::X86::Cpuid::FeatureExtIdEcX[DataBreakpointExtension].
57	Reserved.
56	<b>PerfCtrExtDF</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcX[PerfCtrExtDF]. See Core::X86::Cpuid::FeatureExtIdEcX[PerfCtrExtDF].
55	<b>PerfCtrExtCore</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcX[PerfCtrExtCore]. See Core::X86::Cpuid::FeatureExtIdEcX[PerfCtrExtCore].
54	<b>TopologyExtensions</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcX[TopologyExtensions]. See Core::X86::Cpuid::FeatureExtIdEcX[TopologyExtensions].
53:50	Reserved.
49	<b>TCE</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcX[TCE]. See Core::X86::Cpuid::FeatureExtIdEcX[TCE].
48	<b>FMA4</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcX[FMA4]. See Core::X86::Cpuid::FeatureExtIdEcX[FMA4].
47	<b>LWP</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcX[LWP]. See Core::X86::Cpuid::FeatureExtIdEcX[LWP].
46	Reserved.
45	<b>WDT</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcX[WDT]. See Core::X86::Cpuid::FeatureExtIdEcX[WDT].
44	<b>SKINIT</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcX[SKINIT]. See Core::X86::Cpuid::FeatureExtIdEcX[SKINIT].
43	<b>XOP</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcX[XOP]. See Core::X86::Cpuid::FeatureExtIdEcX[XOP].
42	<b>IBS</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcX[IBS]. See Core::X86::Cpuid::FeatureExtIdEcX[IBS].
41	<b>OSVW</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcX[OSVW]. See Core::X86::Cpuid::FeatureExtIdEcX[OSVW].
40	<b>ThreeDNowPrefetch</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcX[ThreeDNowPrefetch]. See Core::X86::Cpuid::FeatureExtIdEcX[ThreeDNowPrefetch].
39	<b>MisAlignSse</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcX[MisAlignSse]. See Core::X86::Cpuid::FeatureExtIdEcX[MisAlignSse].
38	<b>SSE4A</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcX[SSE4A]. See Core::X86::Cpuid::FeatureExtIdEcX[SSE4A].

37	<b>ABM.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcxC[ABM]. See Core::X86::Cpuid::FeatureExtIdEcxC[ABM].
36	<b>AltMovCr8.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcxC[AltMovCr8]. See Core::X86::Cpuid::FeatureExtIdEcxC[AltMovCr8].
35	<b>ExtApicSpace.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcxC[ExtApicSpace]. See Core::X86::Cpuid::FeatureExtIdEcxC[ExtApicSpace].
34	<b>SVM.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcxC[SVM]. See Core::X86::Cpuid::FeatureExtIdEcxC[SVM].
33	<b>CmpLegacy.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcxC[CmpLegacy]. See Core::X86::Cpuid::FeatureExtIdEcxC[CmpLegacy].
32	<b>LahfSahf.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEcxC[LahfSahf]. See Core::X86::Cpuid::FeatureExtIdEcxC[LahfSahf].
31	<b>ThreeDNow: 3DNow! instructions.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdxC[ThreeDNow]. See Core::X86::Cpuid::FeatureExtIdEdxC[ThreeDNow].
30	<b>ThreeDNowExt.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdxC[ThreeDNowExt]. See Core::X86::Cpuid::FeatureExtIdEdxC[ThreeDNowExt].
29	<b>LM.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdxC[LM]. See Core::X86::Cpuid::FeatureExtIdEdxC[LM].
28	Reserved.
27	<b>RDTSCP.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdxC[RDTSCP]. See Core::X86::Cpuid::FeatureExtIdEdxC[RDTSCP].
26	<b>Page1GB.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdxC[Page1GB]. See Core::X86::Cpuid::FeatureExtIdEdxC[Page1GB].
25	<b>FFXSR.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdxC[FFXSR]. See Core::X86::Cpuid::FeatureExtIdEdxC[FFXSR].
24	<b>FXSR.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdxC[FXSR]. See Core::X86::Cpuid::FeatureExtIdEdxC[FXSR].
23	<b>MMX: MMX instructions.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdxC[MMX]. See Core::X86::Cpuid::FeatureExtIdEdxC[MMX].
22	<b>MmxExt.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdxC[MmxExt]. See Core::X86::Cpuid::FeatureExtIdEdxC[MmxExt].
21	Reserved.
20	<b>NX.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdxC[NX]. See Core::X86::Cpuid::FeatureExtIdEdxC[NX].
19:18	Reserved.
17	<b>PSE36.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdxC[PSE36]. See Core::X86::Cpuid::FeatureExtIdEdxC[PSE36].
16	<b>PAT.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdxC[PAT]. See Core::X86::Cpuid::FeatureExtIdEdxC[PAT].
15	<b>CMOV.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdxC[CMOV]. See Core::X86::Cpuid::FeatureExtIdEdxC[CMOV].
14	<b>MCA.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdxC[MCA]. See Core::X86::Cpuid::FeatureExtIdEdxC[MCA].
13	<b>PGE.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdxC[PGE]. See Core::X86::Cpuid::FeatureExtIdEdxC[PGE].
12	<b>MTRR.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdxC[MTRR]. See Core::X86::Cpuid::FeatureExtIdEdxC[MTRR].
11	<b>SysCallSysRet.</b> Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdxC[SysCallSysRet]. See Core::X86::Cpuid::FeatureExtIdEdxC[SysCallSysRet].
10	Reserved.



9	<b>APIC</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdx[APIC]. See Core::X86::Cpuid::FeatureExtIdEdx[APIC].
8	<b>CMPXCHG8B</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdx[CMPXCHG8B]. See Core::X86::Cpuid::FeatureExtIdEdx[CMPXCHG8B].
7	<b>MCE</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdx[MCE]. See Core::X86::Cpuid::FeatureExtIdEdx[MCE].
6	<b>PAE</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdx[PAE]. See Core::X86::Cpuid::FeatureExtIdEdx[PAE].
5	<b>MSR</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdx[MSR]. See Core::X86::Cpuid::FeatureExtIdEdx[MSR].
4	<b>TSC</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdx[TSC]. See Core::X86::Cpuid::FeatureExtIdEdx[TSC].
3	<b>PSE</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdx[PSE]. See Core::X86::Cpuid::FeatureExtIdEdx[PSE].
2	<b>DE</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdx[DE]. See Core::X86::Cpuid::FeatureExtIdEdx[DE].
1	<b>VME</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdx[VME]. See Core::X86::Cpuid::FeatureExtIdEdx[VME].
0	<b>FPU</b> . Read-write. Reset: Core::X86::Cpuid::FeatureExtIdEdx[FPU]. See Core::X86::Cpuid::FeatureExtIdEdx[FPU].

#### MSRC001\_1019 [Address Mask For DR1 Breakpoint] (Core::X86::Msr::DR1\_ADDR\_MASK)

Read-write. Reset: 0000\_0000\_0000\_0000h.

Support indicated by Core::X86::Cpuid::FeatureExtIdEc[DataBreakpointExtension].

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_1019

Bits	Description
63:32	Reserved.
31:0	<b>AddrMask: mask for DR linear address data breakpoint DR1</b> . Read-write. Reset: 0000_0000h. 1=Exclude bit into address compare. 0=Include bit into address compare. See Core::X86::Msr::DR1_ADDR_MASK. AddrMask[11:0] qualifies the DR1 linear address instruction breakpoint, allowing the DR1 instruction breakpoint on a range of addresses in memory.

#### MSRC001\_101A [Address Mask For DR2 Breakpoint] (Core::X86::Msr::DR2\_ADDR\_MASK)

Read-write. Reset: 0000\_0000\_0000\_0000h.

Support indicated by Core::X86::Cpuid::FeatureExtIdEc[DataBreakpointExtension].

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_101A

Bits	Description
63:32	Reserved.
31:0	<b>AddrMask: mask for DR linear address data breakpoint DR2</b> . Read-write. Reset: 0000_0000h. 1=Exclude bit into address compare. 0=Include bit into address compare. See Core::X86::Msr::DR0_ADDR_MASK. AddrMask[11:0] qualifies the DR2 linear address instruction breakpoint, allowing the DR2 instruction breakpoint on a range of addresses in memory.

#### MSRC001\_101B [Address Mask For DR3 Breakpoint] (Core::X86::Msr::DR3\_ADDR\_MASK)

Read-write. Reset: 0000\_0000\_0000\_0000h.

Support indicated by Core::X86::Cpuid::FeatureExtIdEc[DataBreakpointExtension].

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_101B

Bits	Description
63:32	Reserved.
31:0	<b>AddrMask: mask for DR linear address data breakpoint DR3</b> . Read-write. Reset: 0000_0000h. 1=Exclude bit into address compare. 0=Include bit into address compare. See Core::X86::Msr::DR0_ADDR_MASK. AddrMask[11:0] qualifies the DR3 linear address instruction breakpoint, allowing the DR3 instruction breakpoint on a range of addresses in memory.

**MSRC001\_1027 [Address Mask For DR0 Breakpoints] (Core::X86::Msr::DR0\_ADDR\_MASK)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

Support for DR0[31:12] is indicated by Core::X86::Cpuid::FeatureExtIdEcx[DataBreakpointExtension]. See Core::X86::Msr::DR1\_ADDR\_MASK.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_1027

Bits	Description
63:32	Reserved.
31:0	<b>DR0: mask for DR0 linear address data breakpoint.</b> Read-write. Reset: 0000_0000h. 1=Exclude bit into address compare. 0=Include bit into address compare. See Core::X86::Msr::DR1_ADDR_MASK. This field qualifies the DR0 linear address data breakpoint, allowing the DR0 data breakpoint on a range of addresses in memory. AddrMask[11:0] qualifies the DR0 linear address instruction breakpoint, allowing the DR0 instruction breakpoint on a range of addresses in memory. DR0[31:12] is only valid for data breakpoints. The legacy DR0 breakpoint function is provided by DR0[31:0] == 0000_0000h). The mask bits are active high. DR0 is always used, and it can be used in conjunction with any debug function that uses DR0.

**MSRC001\_1030 [IBS Fetch Control] (Core::X86::Msr::IBS\_FETCH\_CTL)**

Reset: 0000\_0000\_0000\_0000h.

See 2.1.15 [Instruction Based Sampling (IBS)].

The IBS fetch sampling engine is described as follows:

- The periodic fetch counter is an internal 20-bit counter:
  - The periodic fetch counter [19:4] is set to IbsFetchCnt[19:4] and the periodic fetch counter [3:0] is set according to IbsRandEn when IbsFetchEn is changed from 0 to 1.
  - It increments for every fetch cycle that completes when IbsFetchEn == 1 and IbsFetchVal == 0.
    - The periodic fetch counter is undefined when IbsFetchEn == 0 or IbsFetchVal == 1.
  - When IbsFetchCnt[19:4] is read it returns the current value of the periodic fetch counter [19:4].
- When the periodic fetch counter reaches {IbsFetchMaxCnt[19:4],0h} and the selected instruction fetch completes or is aborted:
  - IbsFetchVal is set to 1.
    - Drivers can't assume that IbsFetchCnt[19:4] is 0 when IbsFetchVal == 1.
- The status of the operation is written to the IBS fetch registers (this register, Core::X86::Msr::IBS\_FETCH\_LINADDR and Core::X86::Msr::IBS\_FETCH\_PHYSADDR).
- An interrupt is generated as specified by Core::X86::Msr::IBS\_CTL. The interrupt service routine associated with this interrupt is responsible for saving the performance information stored in IBS execution registers.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_1030

Bits	Description										
63:62	Reserved.										
61	<b>IbsFetchL3Miss: L3 cache miss for the sampled fetch.</b> Read-write, Volatile. Reset: 0. 1=The instruction fetch missed in the L3 Cache on the same CCX.										
60	<b>IbsFetchOcMiss: Op cache miss for the sampled fetch.</b> Read-write, Volatile. Reset: 0. 1=The Op Cache was not able to supply all bytes for the tagged fetch.										
59	<b>IbsL3MissOnly: Only L3 miss samples are reported.</b> Read-write, Volatile. Reset: 0. 1=An IBS interrupt is only created for fetch samples that had an L3 miss; Fetch samples without an L3 miss are discarded and the internal periodic fetch counter is reset to a pseudo random value between 0 and 15. Allows for filtering of Fetch IBS samples based on their L3Miss status.										
58	<b>IbsFetchL2Miss: L2 cache miss for the sampled fetch.</b> Read-write, Volatile. Reset: 0. 1=The instruction fetch missed in the L2 Cache. Qualified by (IbsFetchComp == 1).										
57	<b>IbsRandEn: random instruction fetch tagging enable.</b> Read-write. Reset: 0. 0=Bits[3:0] of the fetch counter are set to 0h when IbsFetchEn is set to start the fetch counter. 1=Bits[3:0] of the fetch counter are randomized when IbsFetchEn is set to start the fetch counter.										
56	<b>IbsL2TlbMiss: instruction cache L2TLB miss.</b> Read-write, Volatile. Reset: 0. 1=The instruction fetch missed in the L2 TLB.										
55	<b>IbsL1TlbMiss: instruction cache L1TLB miss.</b> Read-write, Volatile. Reset: 0. 1=The instruction fetch missed in the L1 TLB.										
54:53	<b>IbsL1TlbPgSz: instruction cache L1TLB page size.</b> Read-write, Volatile. Reset: 0h. Indicates the page size of the translation in the L1 TLB. This field is only valid if IbsPhyAddrValid == 1.										
<b>ValidValues:</b>											
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>4 KB</td></tr> <tr> <td>1h</td><td>2 MB</td></tr> <tr> <td>2h</td><td>1 GB</td></tr> <tr> <td>3h</td><td>16 KB (Coalesced 4 KB pages)</td></tr> </table>	Value	Description	0h	4 KB	1h	2 MB	2h	1 GB	3h	16 KB (Coalesced 4 KB pages)
Value	Description										
0h	4 KB										
1h	2 MB										
2h	1 GB										
3h	16 KB (Coalesced 4 KB pages)										
52	<b>IbsPhyAddrValid: instruction fetch physical address valid.</b> Read-write, Volatile. Reset: 0. 1=The physical address in Core::X86::Msr::IBS_FETCH_PHYSADDR and the IbsL1TlbPgSz field are valid for the instruction fetch.										
51	<b>IbsIcMiss: instruction cache miss.</b> Read-write, Volatile. Reset: 0. 1=The instruction fetch missed in either the instruction cache or the Op Cache.										

50	<b>IbsFetchComp: instruction fetch complete.</b> Read-write, Volatile. Reset: 0. 1=The instruction fetch completed and the data is available for use by the instruction decoder.
49	<b>IbsFetchVal: instruction fetch valid.</b> Read-write, Volatile. Reset: 0. 1=New instruction fetch data available. When this bit is set, the fetch counter stops counting and an interrupt is generated as specified by Core::X86::Msrr::IBS_CTL. This bit must be cleared for the fetch counter to start counting. When clearing this bit, software can write 0000h to IbsFetchCnt[19:4] to start the fetch counter at IbsFetchMaxCnt[19:4].
48	<b>IbsFetchEn: instruction fetch enable.</b> Read-write. Reset: 0. 1=Instruction fetch sampling is enabled.
47:32	<b>IbsFetchLat: instruction fetch latency.</b> Read-write, Volatile. Reset: 0000h. Indicates the number of clock cycles from when the instruction fetch was initiated to when the data was delivered to the core. If the instruction fetch is abandoned before the fetch completes, this field returns the number of clock cycles from when the instruction fetch was initiated to when the fetch was abandoned.
31:16	<b>IbsFetchCnt[19:4].</b> Read-write, Volatile. Reset: 0000h. Provides Read/Write access to bits[19:4] of the periodic fetch counter. Programming this field to a value greater than or equal to IbsFetchMaxCnt[19:4] results in undefined behavior.
15:0	<b>IbsFetchMaxCnt[19:4].</b> Read-write. Reset: 0000h. Specifies bits[19:4] of the maximum count value of the periodic fetch counter. Programming this field to 0000h and setting IbsFetchEn results in undefined behavior. Bits[3:0] of the maximum count are always 0000b.

#### MSRC001\_1031 [IBS Fetch Linear Address] (Core::X86::Msrr::IBS\_FETCH\_LINADDR)

Read-write, Volatile. Reset: 0000\_0000\_0000\_0000h.

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_1031

Bits	Description
63:0	<b>IbsFetchLinAd: instruction fetch linear address.</b> Read-write, Volatile. Reset: 0000_0000_0000_0000h. Provides the linear address in canonical form for the tagged instruction fetch.

#### MSRC001\_1032 [IBS Fetch Physical Address] (Core::X86::Msrr::IBS\_FETCH\_PHYSADDR)

Read-write, Volatile. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_1032

Bits	Description
63:48	Reserved.
47:0	<b>IbsFetchPhysAd: instruction fetch physical address.</b> Read-write, Volatile. Reset: 0000_0000_0000h. Provides the physical address for the tagged instruction fetch. The lower 12 bits are not modified by address translation, so they are always the same as the linear address. This field contains valid data only if Core::X86::Msrr::IBS_FETCH_CTL[IbsPhyAddrValid] is asserted. When nested paging is active, the reported physical address is the system physical address. This register reads zero for a guest with active IBS Virtualization.

**MSRC001\_1033 [IBS Execution Control] (Core::X86::Msr::IBS\_OP\_CTL)**

Reset: 0000\_0000\_0000\_0000h.

See 2.1.15 [Instruction Based Sampling (IBS)].

The IBS execution sampling engine is described as follows for IbsOpCntCtl == 1. If IbsOpCntCtl == 0 then references to "periodic op counter" mean "periodic cycle counter".

- The periodic op counter is an internal 27-bit counter:
  - It is set to IbsOpCurCnt[26:0] when IbsOpEn is changed from 0 to 1.
  - It increments every dispatched macro-op when IbsOpEn == 1 and IbsOpVal == 0.
    - The periodic op counter is undefined when IbsOpEn == 0 or IbsOpVal == 1.
  - When IbsOpCurCnt[26:0] is read then it returns the current value of the periodic op counter [26:0].
- When the periodic op counter reaches IbsOpMaxCnt:
  - The next dispatched op is tagged if IbsOpCntCtl == 1. A valid op in the next dispatched line is tagged if IbsOpCntCtl == 0. See IbsOpCntCtl.
  - The periodic op counter [26:7]=0; [6:0] is randomized by hardware.
- The periodic op counter is not modified when a tagged op is flushed.
- When a tagged op is retired and all sample data has been collected:
  - IbsOpVal is set to 1.
    - Drivers can't assume that IbsOpCurCnt is 0 when IbsOpVal == 1.
- The status of the operation is written to the IBS execution registers (this register, Core::X86::Msr::IBS\_OP\_RIP, Core::X86::Msr::IBS\_OP\_DATA, Core::X86::Msr::IBS\_OP\_DATA2, Core::X86::Msr::IBS\_OP\_DATA3, Core::X86::Msr::IBS\_DC\_LINADDR and Core::X86::Msr::IBS\_DC\_PHYSADDR).
- An interrupt is generated as specified by Core::X86::Msr::IBS\_CTL. The interrupt service routine associated with this interrupt is responsible for saving the performance information stored in IBS execution registers.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_1033

Bits	Description
63	<b>IbsOpLatFltEn.</b> Read-write, Volatile. Reset: 0. Load latency filter enable.
62:59	<b>IbsOpLatThrsh.</b> Read-write, Volatile. Reset: 0h. If latency thresholding is enabled, the latency threshold is calculated as (IbsOpLatThrsh+1) * 128 and compared with the DCIBSDATA.DcMissLatency value.
58:32	<b>IbsOpCurCnt[26:0]: periodic op counter current count.</b> Read-write, Volatile. Reset: 000_0000h. Returns the current value of the periodic op counter.
31:27	Reserved.
26:20	<b>IbsOpMaxCnt[26:20]: periodic op counter maximum count.</b> Read-write. Reset: 00h. See IbsOpMaxCnt[19:4].
19	<b>IbsOpCntCtl: periodic op counter count control.</b> Read-write. Reset: 0. 0=Count clock cycles; a 1-of-4 round-robin counter selects an op in the next dispatch line; if the op pointed to by the round-robin counter is invalid, then the next younger valid op is selected. 1=Count dispatched ops; when a roll-over occurs, the counter is preloaded with a pseudorandom 7 bit value between 1 and 127.
18	<b>IbsOpVal: op sample valid.</b> Read-write, Volatile. Reset: 0. 1=New instruction execution data available; the periodic op counter is disabled from counting. An interrupt may be generated when this bit is set as specified by Core::X86::Msr::IBS_CTL[LvtOffset].
17	<b>IbsOpEn: op sampling enable.</b> Read-write. Reset: 0. 1=Instruction execution sampling enabled.
16	<b>IbsOpL3MissOnly.</b> Read-write. Reset: 0. 1=An IBS interrupt is only created for op samples that had an L3 miss. L3 miss in this context means the data came outside this core's CCX. A hit in another CCX's L3 is considered an L3 miss. Op samples that do not have an L3 miss are dropped and IbsOpCurCnt is reset to a pseudo random value between 0 and 127.
15:0	<b>IbsOpMaxCnt[19:4]: periodic op counter maximum count.</b> Read-write. Reset: 0000h. IbsOpMaxCnt[26:0] = {IbsOpMaxCnt[26:20], IbsOpMaxCnt[19:4], 0000b}. Specifies maximum count value of the periodic op counter. Bits[3:0] of the maximum count are always 0000b.

	<b>Valid Values:</b>	
	<b>Value</b>	<b>Description</b>
	0008h-0000h	Reserved.
	FFFFh-0009h	<Value> *16 Ops.

#### MSRC001\_1034 [IBS Op RIP] (Core::X86::Msr::IBS\_OP\_RIP)

Read-write, Volatile. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_1034

Bits	Description
63:0	<b>IbsOpRip.</b> Read-write, Volatile. Reset: 0000_0000_0000_0000h. 64 bit Segment offset (RIP) of the instruction that contains the tagged op.

#### MSRC001\_1035 [IBS Op Data] (Core::X86::Msr::IBS\_OP\_DATA)

Read-write, Volatile. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_1035

Bits	Description
63:41	Reserved.
40	<b>IbsOpMicrocode.</b> Read-write, Volatile. Reset: 0. 1=Tagged operation from microcode.
39	<b>IbsOpFuse: fused instruction op.</b> Read-write, Volatile. Reset: 0. 1=Tagged operation was part of a fused instruction pair; IBS_OP_RIP reports the rIP of the older instruction within that pair. Support indicated by Core::X86::Cpuid::IbsIdEax[OpFuse].
38	<b>IbsRipInvalid: RIP is invalid.</b> Read-write, Volatile. Reset: 0. 1=Tagged operation RIP is invalid. Support indicated by Core::X86::Cpuid::IbsIdEax[RipInvalidChk].
37	<b>IbsOpBrnRet: branch op retired.</b> Read-write, Volatile. Reset: 0. 1=Tagged operation was a branch op that retired.
36	<b>IbsOpBrnMisp: mispredicted branch op.</b> Read-write, Volatile. Reset: 0. 1=Tagged operation was a branch op that was mispredicted. Qualified by IbsOpBrnRet == 1.
35	<b>IbsOpBrnTaken: taken branch op.</b> Read-write, Volatile. Reset: 0. 1=Tagged operation was a branch op that was taken. Qualified by IbsOpBrnRet == 1.
34	<b>IbsOpReturn: return op.</b> Read-write, Volatile. Reset: 0. 1=Tagged operation was return op. Qualified by (IbsOpBrnRet == 1).
33:32	Reserved.
31:16	<b>IbsTagToRetCtr: op tag to retire count.</b> Read-write, Volatile. Reset: 0000h. This field returns the number of cycles from when the op was tagged to when the op was retired. This field is equal to IbsCompToRetCtr when the tagged op has zero-cycle latency.
15:0	<b>IbsCompToRetCtr: op completion to retire count.</b> Read-write, Volatile. Reset: 0000h. This field returns the number of cycles from when the op was completed to when the op was retired.

**MSRC001\_1036 [IBS Op Data 2] (Core::X86::Msr::IBS\_OP\_DATA2)**

Reset: 0000\_0000\_0000\_0000h.

Data is only valid for load operations that miss both the L1 data cache and the L2 cache.

If a load or store operation crosses a 64B boundary, the data returned in this register is for the lower of the two cache lines accessed.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_1036

Bits	Description																										
63:8	Reserved.																										
7:6	<b>DataSrc[4:3]: Northbridge IBS request data source high bits.</b> Read-write, Volatile. Reset: 0h. High order bits for the DataSrc field. See DataSrcLo for a description of the valid values.																										
5	<b>NearFarCache_NearFar: Requests that return from any cache.</b> Read-write, Volatile. Reset: 0. 0=M State. 1=O State. Valid when the data source type is a cache.																										
4	<b>FarAny_NearFar: Requests that return from another NUMA node.</b> Read-write, Volatile. Reset: 0. 0=The request is serviced by the Northbridge in the same node as the requesting core. 1=The request is serviced by the Northbridge in a different NUMA node than the requesting core. Valid when DataSrc is non-zero.																										
3	Reserved.																										
2:0	<b>DataSrc[2:0]: Northbridge IBS request data source low bits.</b> Read-write. Reset: 0h. <b>Description:</b> Valid Values for {DataSrc[4:3], DataSrc[2:0]: }																										
	<table> <tr> <th>Values</th><th>Description</th></tr> <tr> <td>0h</td><td>No valid status.</td></tr> <tr> <td>1h</td><td>LocalCcx. Local L3 or different L2 in the same CCX.</td></tr> <tr> <td>2h</td><td>NearFarCache_Near. Data belonging to the local NUMA node returned from cache of a different CCX.</td></tr> <tr> <td>3h</td><td>DramIO_Near. Data returned from local node's DRAM/MMIO.</td></tr> <tr> <td>4h</td><td>Reserved.</td></tr> <tr> <td>5h</td><td>NearFarCache_Far. Data belonging to a different NUMA node returned from cache of a different CCX.</td></tr> <tr> <td>6h</td><td>LongLat_NearFar. Long-latency DIMM.</td></tr> <tr> <td>7h</td><td>DramIO_Far. Data returned from different node's DRAM/MMIO.</td></tr> <tr> <td>8h</td><td>Ext_NearFar. Extension Memory.</td></tr> <tr> <td>9h-Bh</td><td>Reserved.</td></tr> <tr> <td>Ch</td><td>Peer_NearFar. Coherent Memory of a different processor type.</td></tr> <tr> <td>Dh-1Fh</td><td>Reserved.</td></tr> </table>	Values	Description	0h	No valid status.	1h	LocalCcx. Local L3 or different L2 in the same CCX.	2h	NearFarCache_Near. Data belonging to the local NUMA node returned from cache of a different CCX.	3h	DramIO_Near. Data returned from local node's DRAM/MMIO.	4h	Reserved.	5h	NearFarCache_Far. Data belonging to a different NUMA node returned from cache of a different CCX.	6h	LongLat_NearFar. Long-latency DIMM.	7h	DramIO_Far. Data returned from different node's DRAM/MMIO.	8h	Ext_NearFar. Extension Memory.	9h-Bh	Reserved.	Ch	Peer_NearFar. Coherent Memory of a different processor type.	Dh-1Fh	Reserved.
Values	Description																										
0h	No valid status.																										
1h	LocalCcx. Local L3 or different L2 in the same CCX.																										
2h	NearFarCache_Near. Data belonging to the local NUMA node returned from cache of a different CCX.																										
3h	DramIO_Near. Data returned from local node's DRAM/MMIO.																										
4h	Reserved.																										
5h	NearFarCache_Far. Data belonging to a different NUMA node returned from cache of a different CCX.																										
6h	LongLat_NearFar. Long-latency DIMM.																										
7h	DramIO_Far. Data returned from different node's DRAM/MMIO.																										
8h	Ext_NearFar. Extension Memory.																										
9h-Bh	Reserved.																										
Ch	Peer_NearFar. Coherent Memory of a different processor type.																										
Dh-1Fh	Reserved.																										



**MSRC001\_1037 [IBS Op Data 3] (Core::X86::Msr::IBS\_OP\_DATA3)**

Read-write, Volatile. Reset: 0000\_0000\_0000\_0000h.

Data in this register is only valid when either Core::X86::Msr::IBS\_OP\_DATA3[IbsStOp] or Core::X86::Msr::IBS\_OP\_DATA3[IbsLdOp] are set.

If a load or store operation crosses a 64B boundary, the data returned in this register is for the lower of the two cache lines accessed.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_1037

Bits	Description																				
63:48	<b>IbsTlbRefillLat: L1 DTLB refill latency.</b> Read-write, Volatile. Reset: 0000h. The number of cycles from when a L1 DTLB refill is triggered by a tagged op to when the L1 DTLB fill has been completed. This field is only valid when Core::X86::Msr::IBS_OP_DATA3[IbsDcPhyAddrValid] is set.																				
47:32	<b>IbsDcMissLat: data cache miss latency.</b> Read-write, Volatile. Reset: 0000h. Indicates the number of clock cycles from when a miss is detected in the data cache to when the data was delivered to the core. The value returned by this counter is not valid for non-load ops (Core::X86::Msr::IBS_OP_DATA3[IbsLdOp]=0) or software prefetch ops (Core::X86::Msr::IBS_OP_DATA3[IbsSwPf]=1).																				
31:26	<b>IbsOpDcMissOpenMemReqs: outstanding memory requests on DC fill.</b> Read-write, Volatile. Reset: 00h. The number of allocated, valid DC MABs when the MAB corresponding to a tagged DC miss op is deallocated. Includes the MAB allocated by the sampled op. 00000b=No information provided.																				
25:22	<b>IbsOpMemWidth: load/store size in bytes.</b> Read-write, Volatile. Reset: 0h. Report the number of bytes the load or store is attempting to access. <b>ValidValues:</b>																				
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>No information provided.</td></tr> <tr> <td>1h</td><td>Byte.</td></tr> <tr> <td>2h</td><td>Word.</td></tr> <tr> <td>3h</td><td>DW (4 Bytes).</td></tr> <tr> <td>4h</td><td>QW (8 Bytes).</td></tr> <tr> <td>5h</td><td>OW (16 Bytes).</td></tr> <tr> <td>6h</td><td>32 Bytes.</td></tr> <tr> <td>7h</td><td>64 Bytes.</td></tr> <tr> <td>Fh-8h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	No information provided.	1h	Byte.	2h	Word.	3h	DW (4 Bytes).	4h	QW (8 Bytes).	5h	OW (16 Bytes).	6h	32 Bytes.	7h	64 Bytes.	Fh-8h	Reserved.
Value	Description																				
0h	No information provided.																				
1h	Byte.																				
2h	Word.																				
3h	DW (4 Bytes).																				
4h	QW (8 Bytes).																				
5h	OW (16 Bytes).																				
6h	32 Bytes.																				
7h	64 Bytes.																				
Fh-8h	Reserved.																				
21	<b>IbsSwPf: software prefetch.</b> Read-write, Volatile. Reset: 0. 1=The op is a software prefetch.																				
20	<b>IbsL2Miss: L2 cache miss for the sampled operation.</b> Read-write, Volatile. Reset: 0. 1=The operation missed in the L2, regardless of whether the op initiated the request to the L2. This is not expected to be set for store operations to non-cacheable memory.																				
19	Reserved.																				
18	<b>IbsDcPhyAddrValid: data cache physical address valid.</b> Read-write, Volatile. Reset: 0. 1=The physical address in Core::X86::Msr::IBS_DC_PHYSADDR is valid for the load or store operation.																				
17	<b>IbsDcLinAddrValid: data cache linear address valid.</b> Read-write, Volatile. Reset: 0. 1=The linear address in Core::X86::Msr::IBS_DC_LINADDR is valid for the load or store operation.																				
16	<b>DcMissNoMabAlloc: DC miss with no MAB allocated.</b> Read-write, Volatile. Reset: 0. 1=The tagged load or store operation hit on an already allocated MAB.																				
15	<b>IbsDcLockedOp: locked operation.</b> Read-write, Volatile. Reset: 0. 1=Tagged load or store operation is a locked operation.																				
14	<b>IbsDcUcMemAcc: UC memory access.</b> Read-write, Volatile. Reset: 0. 1=Tagged load or store operation accessed uncacheable memory.																				
13	<b>IbsDcWcMemAcc: WC memory access.</b> Read-write, Volatile. Reset: 0. 1=Tagged load or store operation accessed write combining memory.																				
12:9	Reserved.																				



8	<b>IbsDcMisAcc: misaligned access.</b> Read-write, Volatile. Reset: 0. 1=The tagged load or store operation crosses a 64 byte address boundary.
7	<b>IbsDcMiss: data cache miss.</b> Read-write, Volatile. Reset: 0. 1=The cache line used by the tagged load or store was not present in the data cache. This is not expected to be set for store operations to non-cacheable memory.
6	Reserved.
5	<b>IbsDcL1TlbHit1G: data cache L1TLB hit in 1G page.</b> Read-write, Volatile. Reset: 0. 1=The physical address for the tagged load or store operation was present in a 1G page table entry in the data cache L1TLB.
4	<b>IbsDcL1TlbHit2M: data cache L1TLB hit in 2M page.</b> Read-write, Volatile. Reset: 0. 1=The physical address for the tagged load or store operation was present in a 2M page table entry in the data cache L1TLB.
3	<b>IbsDcL2TlbMiss: data cache L2TLB miss.</b> Read-write, Volatile. Reset: 0. 1=The physical address for the tagged load or store operation was not present in the data cache L2TLB.
2	<b>IbsDcL1tlbMiss: data cache L1TLB miss.</b> Read-write, Volatile. Reset: 0. 1=The physical address for the tagged load or store operation was not present in the data cache L1TLB.
1	<b>IbsStOp: store op.</b> Read-write, Volatile. Reset: 0. 1=Tagged operation is a store operation.
0	<b>IbsLdOp: load op.</b> Read-write, Volatile. Reset: 0. 1=Tagged operation is a load operation.

**MSRC001\_1038 [IBS DC Linear Address] (Core::X86::Msrr::IBS\_DC\_LINADDR)**

Read-write, Volatile. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_1038

Bits	Description
63:0	<b>IbsDcLinAd.</b> Read-write, Volatile. Reset: 0000_0000_0000_0000h. Provides the linear address in canonical form for the tagged load or store operation. This field contains valid data only if Core::X86::Msrr::IBS_OP_DATA3[IbsDcLinAddrValid] is asserted.

**MSRC001\_1039 [IBS DC Physical Address] (Core::X86::Msrr::IBS\_DC\_PHYSADDR)**

Read-write, Volatile. Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_1039

Bits	Description
63:48	Reserved.
47:0	<b>IbsDcPhysAd: load or store physical address.</b> Read-write, Volatile. Reset: 0000_0000_0000h. Provides the physical address for the tagged load or store operation. The lower 12 bits are not modified by address translation, so they are always the same as the linear address. For memory accesses, that cross a 64B boundary (Core::X86::Msrr::IBS_OP_DATA3[IbsDcMisAcc] is set), this field always points to the first cache line for the access. Cache miss related information in Core::X86::Msrr::IBS_OP_DATA3 or Core::X86::Msrr::IBS_OP_DATA2 will be for one of the two cache lines accessed by the load or store operation. This field contains valid data only if Core::X86::Msrr::IBS_OP_DATA3[IbsDcPhyAddrValid]=1 and Core::X86::Msrr::IBS_OP_DATA3[IbsDcLinAddrValid]=1. When nested paging is active, the reported physical address is the system physical address. This register reads zero for a guest with active IBS Virtualization.

**MSRC001\_103A [IBS Control] (Core::X86::Msrr::IBS\_CTL)**

Read, Error-on-write.

\_lthree[1:0]\_core[7:0]\_thread[1:0]; MSRC001\_103A

Bits	Description
63:9	Reserved.
8	<b>LvtOffsetVal: IBS Fetch and Op local vector table offset valid.</b> Read, Error-on-write. Reset: X.
7:4	Reserved.
3:0	<b>LvtOffset: IBS Fetch and Op local vector table offset.</b> Read, Error-on-write. Reset: Xh.

<b>MSRC001_103B [IBS Branch Target Address] (Core::X86::Msr::BP_IBSTGT_RIP)</b>	
Read-write, Volatile. Reset: 0000_0000_0000_0000h.	
Support for this register indicated by Core::X86::Cpuid::IbsIdEax[BrnTrgt].	
_lthree[1:0]_core[7:0]_thread[1:0]; MSRC001_103B	
Bits	Description
63:0	<b>IbsBrTarget.</b> Read-write, Volatile. Reset: 0000_0000_0000_0000h. The logical address in canonical form for the branch target. It only contains a valid value for branch instructions (Core::X86::Msr::IBS_OP_DATA[IbsOpBrnRet] == 1).
<b>MSRC001_103C [IBS Fetch Control Extended] (Core::X86::Msr::IC_IBS_EXTD_CTL)</b>	
Read-write, Volatile. Reset: 0000_0000_0000_0000h.	
Support for this register indicated by Core::X86::Cpuid::IbsIdEax[IbsFetchCtlExtd].	
_lthree[1:0]_core[7:0]_thread[1:0]; MSRC001_103C	
Bits	Description
63:16	Reserved.
15:0	<b>IbsItlbRefillLat: ITLB Refill Latency for the sampled fetch, if there is a reload.</b> Read-write, Volatile. Reset: 0000h. The number of cycles when the fetch engine is stalled for an ITLB reload for the sampled fetch. If there is no reload, the latency is 0.

### 2.1.13.5 L3 MSRs - MSRC001\_1xxx

See 2.2.1 [L3 MSR Registers].

## 2.1.14 Performance Monitor Counters

### 2.1.14.1 RDPMC Assignments

There are six core performance event counters per thread, six performance events counters per L3 complex and sixteen Data Fabric performance events counters mapped to the RDPMC instruction as follows:

- The RDPMC[5:0] instruction accesses core events. See 2.1.14.5 [Core Performance Monitor Counters].
- The RDPMC[9:6, 1B:10] instruction accesses data fabric events.
- The RDPMC[F:A] instruction accesses L3 cache events. See 2.1.14.6 [L3 Cache Performance Monitor Counters].

### 2.1.14.2 Performance Measurement

This section contains AMD's recommended method for collecting microarchitecture performance common to software optimization. This may require combining multiple performance event selections. Table 18 [Guidance for Common Performance Statistics with Complex Event Selects] lists formulas for collecting common performance statistics.

- The term Event is the full value written to Core::X86::Msr::PERF\_CTL0..5.
  - Core PMC select bits [63:36, 31:16] are at the user's discretion, (i.e., they are not part of the event selection).
- The term L3Event is the full value written to Core::X86::Msr::ChL3PmcCfg.
- The term DFEvent is the full value written to Core::X86::Msr::DF\_PERF\_CTL.

Some UnitMask fields are not disclosed, but may be used by 2.1.14.2 [Performance Measurement].

*Table 18: Guidance for Common Performance Statistics with Complex Event Selects*

Description	Equation
Branch Prediction	

Execution-Time Branch Misprediction Ratio (Non-Speculative).	Event[0x4300C3] / Event[0x4300C2]
Basic Caching	
All Data Cache Accesses	Event[0x430729]
All L2 Cache Accesses	Event[0x43F160] + Event[0x431F70] + Event[0x431F71] + Event[0x431F72]
L2 Cache Access from L1 Instruction Cache Miss (including prefetch)	Event[0x431060]
L2 Cache Access from L1 Data Cache Miss (including Prefetch)	Event[0x43E060]
L2 Cache Access from L2 Cache HWPF	Event[0x431F70] + Event[0x431F71] + Event[0x431F72]
All L2 Cache Misses	Event[0x430964] + Event[0x431F71] + Event[0x431F72]
L2 Cache Miss from L1 Instruction Cache Miss	Event[0x430164]
L2 Cache Miss from L1 Data Cache Miss	Event[0x430864]
L2 Cache Miss from L2 Cache HWPF	Event[0x431F71] + Event[0x431F72]
All L2 Cache Hits	Event[0x43f664] + Event[0x431f70]
L2 Cache Hit from L1 Instruction Cache Miss	Event[0x430664]
L2 Cache Hit from L1 Data Cache Miss	Event[0x43F064]
L2 Cache Hit from L2 Cache HWPF	Event[0x431F70]
L3 Cache Accesses	L3Event[0x0300C0000040FF04]
L3 Miss (includes cacheline state change requests)	L3Event[0x0300C00000400104]
Average L3 Cache Read Miss Latency (in nanoseconds)	L3Event[0x0303C00000403FAC]*10/ L3Event[0x0303C00000403FAD]
Op Cache (64B) Fetch Miss Ratio	Event[0x20043048F] / Event[0x20043078F]
Instruction Cache (32B) Fetch Miss Ratio	Event[0x10043188E] / Event[0x100431F8E]
Advanced Caching	
L1 Data Cache Fills from DRAM or IO in any NUMA node	Event[0x434844]
L1 Data Cache Fills from a different NUMA node	Event[0x435044]
L1 Data Cache Fills from within the same CCX	Event[0x430344]
L1 Data Cache Fills from another CCX cache in any NUMA node	Event[0x431444]
L1 Data Cache Fills All	Event[0x435F44]
Demand L1 Data Cache Fills from local L2	Event[0x430143]
Demand L1 Data Cache Fills from local L3 or different L2 in same CCX	Event[0x430243]
Demand L1 Data Cache Fills from another CCX cache in the same NUMA node	Event[0x430443]
Demand L1 Data Cache Fills from DRAM or MMIO in the same NUMA node	Event[0x430843]
Demand L1 Data Cache Fills from another CCX cache in a different NUMA node	Event[0x431043]
Demand L1 Data Cache Fills from Remote Memory or IO	Event[0x434043]
64B lines written per WCB close	Event[0x430150] / Event[0x432063]
TLBs	
L1 ITLB Misses	Event[0x430084] + Event[0x430785]
L2 ITLB Misses & Instruction page walk	Event[0x430785]

L1 DTLB Misses	Event[0x43FF45]
L2 DTLB Misses & Data page walk	Event[0x43F045]
All TLBs Flushed	Event[0x43FF78]
Stalls	
Macro-ops Dispatched	Event[0x4307AA]
Mixed SSE/AVX Stalls	Event[0x430E0E]
Macro-ops Retired	Event[0x4300C1]

### 2.1.14.3 Pipeline Utilization Analysis

Table 19: Guidance for Pipeline Utilization Analysis Statistics

Name	Description	Equation
Level 1		
Total Dispatch Slots	Up to 8 instructions can be dispatched in one cycle.	$8 * \text{Event}[430076]$
Frontend Bound	Fraction of dispatch slots that remained unused because the frontend did not supply enough instructions/ops.	$\text{Event}[1004301A0] / \text{Total Dispatch Slots}$
Bad Speculation	Fraction of dispatched ops that did not retire.	$(\text{Event}[4307AA] - \text{Event}[4300C1]) / \text{Total Dispatch Slots}$
Backend Bound	Fraction of dispatch slots that remained unused because of backend stalls.	$\text{Event}[100431EA0] / \text{Total Dispatch Slots}$
SMT contention	Fraction of unused dispatch slots because the other thread was selected.	$\text{Event}[1004360A0] / \text{Total Dispatch Slots}$
Retiring	Fraction of dispatch slots used by ops that retired.	$\text{Event}[4300C1] / \text{Total Dispatch Slots}$
Level 2		
Frontend Bound - Latency	Fraction of dispatch slots that remained unused because of a latency bottleneck in the frontend, such as Instruction Cache or ITLB misses.	$8 * \text{Event}[1084301A0] / \text{Total Dispatch Slots}$
Frontend Bound - BW	Fraction of dispatch slots that remained unused because of a bandwidth bottleneck in the frontend, such as decode bandwidth or Op Cache fetch bandwidth.	$\text{Event}[1004301A0] - (8 * \text{Event}[1084301A0]) / \text{Total Dispatch Slots}$
Bad Speculation – Mispredicts	Fraction of dispatched ops that were flushed due to branch mispredicts.	$\text{Bad Speculation} * \text{Event}[4300C3] / (\text{Event}[4300C3] + \text{Event}[43019F])$
Bad Speculation - Pipeline Restarts	Fraction of dispatched ops that were flushed due to pipeline restarts (resyncs).	$\text{Bad Speculation} * \text{Event}[43019F] / (\text{Event}[4300C3] + \text{Event}[430796])$
Backend Bound - Memory	Fraction of dispatched slots that remained unused because of stalls due to the memory subsystem.	$\text{Backend Bound} * (\text{Event}[43A2D6] / \text{Event}[4302D6])$
Backend Bound – CPU	Fraction of dispatched slots that remained unused because of stalls not related to the memory subsystem.	$\text{Backend Bound} * (1 - (\text{Event}[43A2D6] / \text{Event}[4302D6]))$

Retiring - Fastpath	Fraction of dispatch slots used by fastpath ops that retired.	$\text{Retiring} * (\text{Event}[4300C1] - \text{Event}[1004300C2]) / \text{Event}[4300C1]$
Retiring - Microcode	Fraction of dispatch slots used by microcode ops that retired.	$\text{Retiring} * \text{Event}[1004300C2] / \text{Event}[4300C1]$

#### 2.1.14.4 Large Increment per Cycle Events

Table 20: PMC\_Definitions

Term	Description
<b>MergeEvent</b>	A PMC event that is capable of counter increments greater than 15, thus requiring merging a pair of even/odd performance monitors.

The maximum increment for a regular performance event is 15 (i.e., a 4-bit event). However some event types can have a larger increments every cycle.

An option is provided for merging a pair of even/odd performance monitors to acquire an accurate count. First the odd numbered Core::X86::Msr::PERF\_CTL0..5 is programmed with the event Core::X86::Pmc::Core::Merge (PMCxFFF) with the enable bit (En) turned on and with the remaining bits off. Then the corresponding even numbered Core::X86::Msr::PERF\_CTL0..5 is programmed with the desired PMC event. Both the odd and even numbered counter need to be enabled in Core::X86::Msr::PerfCntrGlobalCtl for the merged counter to count. The performance monitor combines the count value to an 8-bit increment event and extends the counter to a 64-bit counter.

Software wanting to preload a value to a merged counter pair writes the high-order 16-bit value to the low-order 16 bits of the odd counter and then writes the low-order 48-bit value to the even counter. Reading the even counter of the merged counter pair returns the full 64-bit value.

If an even performance monitor is programmed with the event Core::X86::Pmc::Core::Merge the Read results are undetermined. If an even performance monitor is programmed with a non-merge-able event (i.e., a 4-bit event) while the corresponding odd performance monitor is programmed as Merge, the Read results are undetermined. When discontinuing use of a merged counter pair, clear the Merge event from the odd performance monitor.

#### 2.1.14.5 Core Performance Monitor Counters

This section provides the core performance counter events that may be selected through Core::X86::Msr::PERF\_CTL0[EventSelect[11:8],EventSelect[7:0],UnitMask]. See Core::X86::Msr::PERF\_CTR. See Core::X86::Msr::PERF\_LEGACY\_CTL0..3 and Core::X86::Msr::PERF\_LEGACY\_CTR.

## 2.1.14.5.1 Floating-Point (FP) Events

PMCx002 [FP retired x87 uops] (Core::X86::Pmc::Core::Retired_x87_FP_Ops)		
Read-write.		
Number of retired x87 arithmetic operations. Can be used to calculate x87 FLOPs.		
PMCx002		
Bits	Description	
7:3	Reserved.	
2	<b>DivSqrROps.</b> Read-write. x87 Divide or square root uops.	
1	<b>MulOps.</b> Read-write. x87 Multiply uops.	
0	<b>AddSubOps.</b> Read-write. x87 Add/subtract uops.	

PMCx003 [FP retired SSE and AVX FLOPs] (Core::X86::Pmc::Core::Retired_SSE_AVX_FLOPs)		
Read-write.		
Number of SSE and AVX floating point arithmetic operations retired. Number of arithmetic operations retired is dependent on number of uops retired, data size (scalar/128/256/512), data type (BF16/FP16/FP32/FP64) and type of operation (add/sub/mul/mac/...). Use MergeEvent feature for accurate results.		
PMCx003		
Bits	Description	
7:5	<b>FlopTypeSel.</b> Read-write. Mask for specifying FLOP type.	
	<b>ValidValues:</b>	
	<b>Value</b>	<b>Description</b>
	0h	All types.
	1h	B Float 16.
	2h	Scalar single.
	3h	Packed single.
	4h	Scalar double.
	5h	Packed double.
	7h-6h	Reserved.
4	Reserved.	
3	<b>MacFLOPs.</b> Read-write. Each MAC operation count as 2 FLOPs. bfloat MAC operations are not included in this event.	
2	<b>DivFLOPs.</b> Read-write. Divide/square root FLOPs. Does not provide a useful count without use of the MergeEvent feature.	
1	<b>MultFLOPs.</b> Read-write. Multiply FLOPs. Does not provide a useful count without use of the MergeEvent feature.	
0	<b>AddSubFLOPs.</b> Read-write. Add/subtract FLOPs. Does not provide a useful count without use of the MergeEvent feature.	

**PMCx008 [FP uops retired by size] (Core::X86::Pmc::Core::Retired\_FP\_uOps)**

Read-write.

Report number of FP uops retired by size. Can be used to determine how vectorized code is and how much MMX / x87 content is in the code.

PMCx008

Bits	Description
7:6	Reserved.
5	<b>Pack512uOpsRetired.</b> Read-write. Packed 512-bit uops retired.
4	<b>Pack256uOpsRetired.</b> Read-write. Packed 256-bit uops retired.
3	<b>Pack128uOpsRetired.</b> Read-write. Packed 128-bit uops retired.
2	<b>ScalaruOpsRetired.</b> Read-write. Scalar uops retired.
1	<b>MMXuOpsRetired.</b> Read-write. MMX uops retired.
0	<b>x87uOpsRetired.</b> Read-write. x87 uops retired.

**PMCx00A [FP uops retired sorted by vector or scalar] (Core::X86::Pmc::Core::FP\_Ops\_Retired)**

Read-write.

Number of FP uops retired of selected type sorted by vector (AVX/SSE packed) or scalar (x87, AVX/SSE scalar). Can be used to profile FP codes.

PMCx00A

Bits	Description																																		
7:4	<b>VectorFpOpType.</b> Read-write. select a vector FP uop type to count or 0 for none. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr><td>0h</td><td>None selected.</td></tr> <tr><td>1h</td><td>Add.</td></tr> <tr><td>2h</td><td>Subtract.</td></tr> <tr><td>3h</td><td>Multiply.</td></tr> <tr><td>4h</td><td>Multiply accumulate.</td></tr> <tr><td>5h</td><td>Divide.</td></tr> <tr><td>6h</td><td>Square root.</td></tr> <tr><td>7h</td><td>Compare.</td></tr> <tr><td>8h</td><td>Convert.</td></tr> <tr><td>9h</td><td>Blend.</td></tr> <tr><td>Ah</td><td>Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.</td></tr> <tr><td>Bh</td><td>Shuffle. Shuf uop counts may count for instructions that are not necessarily thought to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.</td></tr> <tr><td>Ch</td><td>BFloat.</td></tr> <tr><td>Dh</td><td>Logical.</td></tr> <tr><td>Eh</td><td>Other uops not included in previous groups.</td></tr> <tr><td>Fh</td><td>Select all fp type uops.</td></tr> </table>	Value	Description	0h	None selected.	1h	Add.	2h	Subtract.	3h	Multiply.	4h	Multiply accumulate.	5h	Divide.	6h	Square root.	7h	Compare.	8h	Convert.	9h	Blend.	Ah	Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.	Bh	Shuffle. Shuf uop counts may count for instructions that are not necessarily thought to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.	Ch	BFloat.	Dh	Logical.	Eh	Other uops not included in previous groups.	Fh	Select all fp type uops.
Value	Description																																		
0h	None selected.																																		
1h	Add.																																		
2h	Subtract.																																		
3h	Multiply.																																		
4h	Multiply accumulate.																																		
5h	Divide.																																		
6h	Square root.																																		
7h	Compare.																																		
8h	Convert.																																		
9h	Blend.																																		
Ah	Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.																																		
Bh	Shuffle. Shuf uop counts may count for instructions that are not necessarily thought to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.																																		
Ch	BFloat.																																		
Dh	Logical.																																		
Eh	Other uops not included in previous groups.																																		
Fh	Select all fp type uops.																																		
3:0	<b>ScalarFpOpType.</b> Read-write. select scalar FP uop type to count or 0 for none. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr><td>0h</td><td>None selected.</td></tr> <tr><td>1h</td><td>Add.</td></tr> <tr><td>2h</td><td>Subtract.</td></tr> <tr><td>3h</td><td>Multiply.</td></tr> <tr><td>4h</td><td>Multiply accumulate.</td></tr> <tr><td>5h</td><td>Divide.</td></tr> <tr><td>6h</td><td>Square root.</td></tr> <tr><td>7h</td><td>Compare.</td></tr> <tr><td>8h</td><td>Convert.</td></tr> <tr><td>9h</td><td>Blend.</td></tr> <tr><td>Ah</td><td>Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.</td></tr> <tr><td>Bh</td><td>Shuffle. Shuf uop counts may count for instructions that are not necessarily thought to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.</td></tr> <tr><td>Ch</td><td>BFloat.</td></tr> <tr><td>Dh</td><td>Logical.</td></tr> <tr><td>Eh</td><td>Other uops not included in previous groups.</td></tr> <tr><td>Fh</td><td>Select all fp type uops.</td></tr> </table>	Value	Description	0h	None selected.	1h	Add.	2h	Subtract.	3h	Multiply.	4h	Multiply accumulate.	5h	Divide.	6h	Square root.	7h	Compare.	8h	Convert.	9h	Blend.	Ah	Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.	Bh	Shuffle. Shuf uop counts may count for instructions that are not necessarily thought to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.	Ch	BFloat.	Dh	Logical.	Eh	Other uops not included in previous groups.	Fh	Select all fp type uops.
Value	Description																																		
0h	None selected.																																		
1h	Add.																																		
2h	Subtract.																																		
3h	Multiply.																																		
4h	Multiply accumulate.																																		
5h	Divide.																																		
6h	Square root.																																		
7h	Compare.																																		
8h	Convert.																																		
9h	Blend.																																		
Ah	Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.																																		
Bh	Shuffle. Shuf uop counts may count for instructions that are not necessarily thought to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.																																		
Ch	BFloat.																																		
Dh	Logical.																																		
Eh	Other uops not included in previous groups.																																		
Fh	Select all fp type uops.																																		



**PMCx00B [FP executed integer type uops sorted by vector or scalar] (Core::X86::Pmc::Core::INT\_Ops\_Retired)**

Read-write.

Number of integer uops executed in the FP retired of selected type sorted by vector (SSE/AVX) or scalar (MMX). Can be used to profile vector INT / MMX codes.

PMCx00B

Bits	Description																																		
7:4	<b>SseAvxOpType.</b> Read-write. select SSE/AVX vector INT uop type to count or 0 for none. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr><td>0h</td><td>None selected.</td></tr> <tr><td>1h</td><td>Add.</td></tr> <tr><td>2h</td><td>Subtract.</td></tr> <tr><td>3h</td><td>Multiply.</td></tr> <tr><td>4h</td><td>Multiply accumulate.</td></tr> <tr><td>5h</td><td>AES.</td></tr> <tr><td>6h</td><td>SHA.</td></tr> <tr><td>7h</td><td>Compare.</td></tr> <tr><td>8h</td><td>Convert or pack.</td></tr> <tr><td>9h</td><td>Shift or rotate.</td></tr> <tr><td>Ah</td><td>Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.</td></tr> <tr><td>Bh</td><td>Shuffle. Shuf uop counts may count for instructions that are not necessarily though to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.</td></tr> <tr><td>Ch</td><td>VNNI.</td></tr> <tr><td>Dh</td><td>Logical.</td></tr> <tr><td>Eh</td><td>Other uops not included in previous groups.</td></tr> <tr><td>Fh</td><td>Select all int type uops.</td></tr> </table>	Value	Description	0h	None selected.	1h	Add.	2h	Subtract.	3h	Multiply.	4h	Multiply accumulate.	5h	AES.	6h	SHA.	7h	Compare.	8h	Convert or pack.	9h	Shift or rotate.	Ah	Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.	Bh	Shuffle. Shuf uop counts may count for instructions that are not necessarily though to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.	Ch	VNNI.	Dh	Logical.	Eh	Other uops not included in previous groups.	Fh	Select all int type uops.
Value	Description																																		
0h	None selected.																																		
1h	Add.																																		
2h	Subtract.																																		
3h	Multiply.																																		
4h	Multiply accumulate.																																		
5h	AES.																																		
6h	SHA.																																		
7h	Compare.																																		
8h	Convert or pack.																																		
9h	Shift or rotate.																																		
Ah	Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.																																		
Bh	Shuffle. Shuf uop counts may count for instructions that are not necessarily though to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.																																		
Ch	VNNI.																																		
Dh	Logical.																																		
Eh	Other uops not included in previous groups.																																		
Fh	Select all int type uops.																																		
3:0	<b>MmxOpType.</b> Read-write. select MMX INT scalar uop type to count or 0 for none. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr><td>0h</td><td>None selected.</td></tr> <tr><td>1h</td><td>Add.</td></tr> <tr><td>2h</td><td>Subtract.</td></tr> <tr><td>3h</td><td>Multiply.</td></tr> <tr><td>4h</td><td>Multiply accumulate.</td></tr> <tr><td>5h</td><td>AES.</td></tr> <tr><td>6h</td><td>SHA.</td></tr> <tr><td>7h</td><td>Compare.</td></tr> <tr><td>8h</td><td>Convert or pack.</td></tr> <tr><td>9h</td><td>Shift or rotate.</td></tr> <tr><td>Ah</td><td>Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.</td></tr> <tr><td>Bh</td><td>Shuffle. Shuf uop counts may count for instructions that are not necessarily though to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.</td></tr> <tr><td>Ch</td><td>VNNI.</td></tr> <tr><td>Dh</td><td>Logical.</td></tr> <tr><td>Eh</td><td>Other uops not included in previous groups.</td></tr> <tr><td>Fh</td><td>Select all int type uops.</td></tr> </table>	Value	Description	0h	None selected.	1h	Add.	2h	Subtract.	3h	Multiply.	4h	Multiply accumulate.	5h	AES.	6h	SHA.	7h	Compare.	8h	Convert or pack.	9h	Shift or rotate.	Ah	Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.	Bh	Shuffle. Shuf uop counts may count for instructions that are not necessarily though to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.	Ch	VNNI.	Dh	Logical.	Eh	Other uops not included in previous groups.	Fh	Select all int type uops.
Value	Description																																		
0h	None selected.																																		
1h	Add.																																		
2h	Subtract.																																		
3h	Multiply.																																		
4h	Multiply accumulate.																																		
5h	AES.																																		
6h	SHA.																																		
7h	Compare.																																		
8h	Convert or pack.																																		
9h	Shift or rotate.																																		
Ah	Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.																																		
Bh	Shuffle. Shuf uop counts may count for instructions that are not necessarily though to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.																																		
Ch	VNNI.																																		
Dh	Logical.																																		
Eh	Other uops not included in previous groups.																																		
Fh	Select all int type uops.																																		

**PMCx00C [FP uops retired sorted by packed 128 or packed 256]  
(Core::X86::Pmc::Core::Packed\_FP\_Ops\_Retired)**

Read-write.

Number of FP uops retired of selected type sorted by 128-bit packed dest (XMM) or 256-bit packed dest (YMM). Can be used to profile FP codes.

PMCx00C

Bits	Description																																		
7:4	<b>Fp256OpType.</b> Read-write. select a 256-bit packed FP uop type to count or 0 for none. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr><td>0h</td><td>None selected.</td></tr> <tr><td>1h</td><td>Add.</td></tr> <tr><td>2h</td><td>Subtract.</td></tr> <tr><td>3h</td><td>Multiply.</td></tr> <tr><td>4h</td><td>Multiply accumulate.</td></tr> <tr><td>5h</td><td>Divide.</td></tr> <tr><td>6h</td><td>Square root.</td></tr> <tr><td>7h</td><td>Compare.</td></tr> <tr><td>8h</td><td>Convert.</td></tr> <tr><td>9h</td><td>Blend.</td></tr> <tr><td>Ah</td><td>Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.</td></tr> <tr><td>Bh</td><td>Shuffle. Shuf uop counts may count for instructions that are not necessarily thought to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.</td></tr> <tr><td>Ch</td><td>BFloat.</td></tr> <tr><td>Dh</td><td>Logical.</td></tr> <tr><td>Eh</td><td>Other uops not included in previous groups.</td></tr> <tr><td>Fh</td><td>Select all fp type uops.</td></tr> </table>	Value	Description	0h	None selected.	1h	Add.	2h	Subtract.	3h	Multiply.	4h	Multiply accumulate.	5h	Divide.	6h	Square root.	7h	Compare.	8h	Convert.	9h	Blend.	Ah	Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.	Bh	Shuffle. Shuf uop counts may count for instructions that are not necessarily thought to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.	Ch	BFloat.	Dh	Logical.	Eh	Other uops not included in previous groups.	Fh	Select all fp type uops.
Value	Description																																		
0h	None selected.																																		
1h	Add.																																		
2h	Subtract.																																		
3h	Multiply.																																		
4h	Multiply accumulate.																																		
5h	Divide.																																		
6h	Square root.																																		
7h	Compare.																																		
8h	Convert.																																		
9h	Blend.																																		
Ah	Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.																																		
Bh	Shuffle. Shuf uop counts may count for instructions that are not necessarily thought to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.																																		
Ch	BFloat.																																		
Dh	Logical.																																		
Eh	Other uops not included in previous groups.																																		
Fh	Select all fp type uops.																																		
3:0	<b>Fp128OpType.</b> Read-write. select 128-bit packed FP uop type to count or 0 for none. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr><td>0h</td><td>None selected.</td></tr> <tr><td>1h</td><td>Add.</td></tr> <tr><td>2h</td><td>Subtract.</td></tr> <tr><td>3h</td><td>Multiply.</td></tr> <tr><td>4h</td><td>Multiply accumulate.</td></tr> <tr><td>5h</td><td>Divide.</td></tr> <tr><td>6h</td><td>Square root.</td></tr> <tr><td>7h</td><td>Compare.</td></tr> <tr><td>8h</td><td>Convert.</td></tr> <tr><td>9h</td><td>Blend.</td></tr> <tr><td>Ah</td><td>Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.</td></tr> <tr><td>Bh</td><td>Shuffle. Shuf uop counts may count for instructions that are not necessarily thought to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.</td></tr> <tr><td>Ch</td><td>BFloat.</td></tr> <tr><td>Dh</td><td>Logical.</td></tr> <tr><td>Eh</td><td>Other uops not included in previous groups.</td></tr> <tr><td>Fh</td><td>Select all fp type uops.</td></tr> </table>	Value	Description	0h	None selected.	1h	Add.	2h	Subtract.	3h	Multiply.	4h	Multiply accumulate.	5h	Divide.	6h	Square root.	7h	Compare.	8h	Convert.	9h	Blend.	Ah	Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.	Bh	Shuffle. Shuf uop counts may count for instructions that are not necessarily thought to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.	Ch	BFloat.	Dh	Logical.	Eh	Other uops not included in previous groups.	Fh	Select all fp type uops.
Value	Description																																		
0h	None selected.																																		
1h	Add.																																		
2h	Subtract.																																		
3h	Multiply.																																		
4h	Multiply accumulate.																																		
5h	Divide.																																		
6h	Square root.																																		
7h	Compare.																																		
8h	Convert.																																		
9h	Blend.																																		
Ah	Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.																																		
Bh	Shuffle. Shuf uop counts may count for instructions that are not necessarily thought to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.																																		
Ch	BFloat.																																		
Dh	Logical.																																		
Eh	Other uops not included in previous groups.																																		
Fh	Select all fp type uops.																																		

**PMCx00D [FP executed packed integer uops sorted by packed 128 or packed 256]  
(Core::X86::Pmc::Core::Packed\_INT\_Ops\_Retired)**

Read-write.

Number of integer uops executed in FP retired of selected type sorted by 128-bit packed dest (XMM) or 256-bit packed dest (YMM). Can be used to profile FP codes.

PMCx00D

Bits	Description																																		
7:4	<p><b>Int256OpType.</b> Read-write. select a 256-bit packed INT uop type to count or 0 for none.</p> <p><b>ValidValues:</b></p> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>None selected.</td></tr> <tr> <td>1h</td><td>Add.</td></tr> <tr> <td>2h</td><td>Subtract.</td></tr> <tr> <td>3h</td><td>Multiply.</td></tr> <tr> <td>4h</td><td>Multiply accumulate.</td></tr> <tr> <td>5h</td><td>AES.</td></tr> <tr> <td>6h</td><td>SHA.</td></tr> <tr> <td>7h</td><td>Compare.</td></tr> <tr> <td>8h</td><td>Convert or pack.</td></tr> <tr> <td>9h</td><td>Shift or rotate.</td></tr> <tr> <td>Ah</td><td>Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.</td></tr> <tr> <td>Bh</td><td>Shuffle. Shuf uop counts may count for instructions that are not necessarily though to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.</td></tr> <tr> <td>Ch</td><td>VNNI.</td></tr> <tr> <td>Dh</td><td>Logical.</td></tr> <tr> <td>Eh</td><td>Other uops not included in previous groups.</td></tr> <tr> <td>Fh</td><td>Select all int type uops.</td></tr> </table>	Value	Description	0h	None selected.	1h	Add.	2h	Subtract.	3h	Multiply.	4h	Multiply accumulate.	5h	AES.	6h	SHA.	7h	Compare.	8h	Convert or pack.	9h	Shift or rotate.	Ah	Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.	Bh	Shuffle. Shuf uop counts may count for instructions that are not necessarily though to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.	Ch	VNNI.	Dh	Logical.	Eh	Other uops not included in previous groups.	Fh	Select all int type uops.
Value	Description																																		
0h	None selected.																																		
1h	Add.																																		
2h	Subtract.																																		
3h	Multiply.																																		
4h	Multiply accumulate.																																		
5h	AES.																																		
6h	SHA.																																		
7h	Compare.																																		
8h	Convert or pack.																																		
9h	Shift or rotate.																																		
Ah	Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.																																		
Bh	Shuffle. Shuf uop counts may count for instructions that are not necessarily though to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.																																		
Ch	VNNI.																																		
Dh	Logical.																																		
Eh	Other uops not included in previous groups.																																		
Fh	Select all int type uops.																																		
3:0	<p><b>Int128OpType.</b> Read-write. select 128-bit packed INT uop type to count or 0 for none.</p> <p><b>ValidValues:</b></p> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>None selected.</td></tr> <tr> <td>1h</td><td>Add.</td></tr> <tr> <td>2h</td><td>Subtract.</td></tr> <tr> <td>3h</td><td>Multiply.</td></tr> <tr> <td>4h</td><td>Multiply accumulate.</td></tr> <tr> <td>5h</td><td>AES.</td></tr> <tr> <td>6h</td><td>SHA.</td></tr> <tr> <td>7h</td><td>Compare.</td></tr> <tr> <td>8h</td><td>Convert or pack.</td></tr> <tr> <td>9h</td><td>Shift or rotate.</td></tr> <tr> <td>Ah</td><td>Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.</td></tr> <tr> <td>Bh</td><td>Shuffle. Shuf uop counts may count for instructions that are not necessarily though to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.</td></tr> <tr> <td>Ch</td><td>VNNI.</td></tr> <tr> <td>Dh</td><td>Logical.</td></tr> <tr> <td>Eh</td><td>Other uops not included in previous groups.</td></tr> <tr> <td>Fh</td><td>Select all int type uops.</td></tr> </table>	Value	Description	0h	None selected.	1h	Add.	2h	Subtract.	3h	Multiply.	4h	Multiply accumulate.	5h	AES.	6h	SHA.	7h	Compare.	8h	Convert or pack.	9h	Shift or rotate.	Ah	Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.	Bh	Shuffle. Shuf uop counts may count for instructions that are not necessarily though to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.	Ch	VNNI.	Dh	Logical.	Eh	Other uops not included in previous groups.	Fh	Select all int type uops.
Value	Description																																		
0h	None selected.																																		
1h	Add.																																		
2h	Subtract.																																		
3h	Multiply.																																		
4h	Multiply accumulate.																																		
5h	AES.																																		
6h	SHA.																																		
7h	Compare.																																		
8h	Convert or pack.																																		
9h	Shift or rotate.																																		
Ah	Move. MOV* instructions will count as INT type, not FP type. In other words, PMCx00A, PMCx00C will not count MOV ops.																																		
Bh	Shuffle. Shuf uop counts may count for instructions that are not necessarily though to include shuffles. i.e. horizontal add, dot-product, and some MOV instructions.																																		
Ch	VNNI.																																		
Dh	Logical.																																		
Eh	Other uops not included in previous groups.																																		
Fh	Select all int type uops.																																		

**PMCx00E [FP Dispatch Faults] (Core::X86::Pmc::Core::FP\_Dispatch\_Faults)**

Read-write.

Number of FP dispatch faults triggered by type. Dispatch fill/spill faults occur when FP either does not have the data needed to operate on in its local registers (fill), or FP needs to empty out upper register data for proper SSE merging behavior when executing AVX code (spill).

PMCx00E

Bits	Description
7:4	Reserved.
3	<b>YmmSpillFault.</b> Read-write. YMM spill fault
2	<b>YmmFillFault.</b> Read-write. YMM fill fault
1	<b>XmmFillFault.</b> Read-write. XMM Fill fault
0	<b>x87FillFault.</b> Read-write. x87 Fill fault

**2.1.14.5.2 Load/Store (LS) Events****PMCx024 [Bad Status 2] (Core::X86::Pmc::Core::Bad\_Status\_2\_STLI)**

Read-write.

Store To Load Interlock (STLI) are loads that were unable to complete because of a possible match with an older store, and the older store could not do Store To Load Forwarding (STLF) for some reason.

PMCx024

Bits	Description
7:2	Reserved.
1	<b>StliOther.</b> Read-write. Store-to-load conflicts: A load was unable to complete due to a non-forwardable conflict with an older store. Most commonly, a load's address range partially but not completely overlaps with an uncompleted older store. Software can avoid this problem by using same-size and same-alignment loads and stores when accessing the same data. Vector/SIMD code is particularly susceptible to this problem; software should construct wide vector stores by manipulating vector elements in registers using shuffle/blend/swap instructions prior to storing to memory, instead of using narrow element-by-element stores.
0	Reserved.

**PMCx025 [Retired Lock Instructions] (Core::X86::Pmc::Core::Retired\_Lock\_Instructions)**

Read-write.

Counts retired atomic read-modify-write instructions with a LOCK prefix.

PMCx025

Bits	Description
7:5	Reserved.
4:0	<b>LockInstructions.</b> Read-write. Specifies type of lock instructions counted
<b>ValidValues:</b>	
Value	Description
00h	Reserved.
01h	BusLock: Non-cacheable or cacheline-misaligned lock.
1Eh-02h	Reserved.
1Fh	AnyLock: Counts all lock instructions.

**PMCx026 [Retired CLFLUSH Instructions] (Core::X86::Pmc::Core::CLFLUSH)**

Read-write.

The number of retired CLFLUSH instructions. This is a non-speculative event.

PMCx026

**Bits Description**

7:0 Reserved.

**PMCx027 [Retired CPUID Instructions] (Core::X86::Pmc::Core::CUID)**

Read-write.

The number of CPUID instructions retired.

PMCx027

**Bits Description**

7:0 Reserved.

**PMCx029 [LS Dispatch] (Core::X86::Pmc::Core::LS\_Dispatch)**

Read-write.

Counts the number of operations dispatched to the LS unit. Unit Masks events are ADDED.

PMCx029

**Bits Description**

7:3 Reserved.

2 **LdOpSt.** Read-write. Dispatch of a single op that performs a load from and store to the same memory address.1 **PureSt.** Read-write. Dispatch of a single op that performs a memory store.0 **PureLd.** Read-write. Dispatch of a single op that performs a memory load.**PMCx02B [SMIs Received] (Core::X86::Pmc::Core::SMI\_or\_SMM\_cycles)**

Reset: 00h.

Counts the number of System Management Interrupts (SMIs) received.

PMCx02B

**Bits Description**

7:0 Reserved.

**PMCx02C [Interrupts Taken] (Core::X86::Pmc::Core::Interrupts\_Taken)**

Read-write.

Counts the number of interrupts taken.

PMCx02C

**Bits Description**

7:1 Reserved.

0 **NumInterrupts.** Read-write. Number of interrupts taken. This event is also counted when UnitMask[7:0]=0.**PMCx035 [Store to Load Forward] (Core::X86::Pmc::Core::Store\_to\_Load\_Forward)**

Read-write.

Number of STLF hits.

PMCx035

**Bits Description**

7:0 Reserved.

**PMCx037 [Store Globally Visible Cancels 2] (Core::X86::Pmc::Core::Store\_Globally\_Visible\_Cancels\_2)**

Read-write.

Counts reasons why a Store Coalescing Buffer (SCB) commit is canceled.

PMCx037

Bits	Description
7:1	Reserved.
0	<b>OlderStVisibleDepCancel.</b> Read-write. Older SCB we are waiting on to become globally visible was unable to become globally visible.

**PMCx041 [LS MAB Allocates by Type] (Core::X86::Pmc::Core::LS\_MAB\_Allocates\_by\_Type)**

Read-write.

Counts when an LS pipe allocates a Miss Address Buffer (MAB) entry to make a miss request.

PMCx041

Bits	Description														
7	Reserved.														
6:0	<b>LsMabAllocation.</b> Read-write. <b>ValidValues:</b>														
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>06h-00h</td><td>Reserved.</td></tr> <tr> <td>07h</td><td>Load Store Allocations</td></tr> <tr> <td>08h</td><td>Hardware Prefetcher Allocations</td></tr> <tr> <td>0Eh-09h</td><td>Reserved.</td></tr> <tr> <td>0Fh</td><td>All Allocations</td></tr> <tr> <td>7Fh-10h</td><td>Reserved.</td></tr> </table>	Value	Description	06h-00h	Reserved.	07h	Load Store Allocations	08h	Hardware Prefetcher Allocations	0Eh-09h	Reserved.	0Fh	All Allocations	7Fh-10h	Reserved.
Value	Description														
06h-00h	Reserved.														
07h	Load Store Allocations														
08h	Hardware Prefetcher Allocations														
0Eh-09h	Reserved.														
0Fh	All Allocations														
7Fh-10h	Reserved.														

**PMCx043 [Demand Data Cache Fills by Data Source] (Core::X86::Pmc::Core::Demand\_DC\_Fills\_by\_Data\_Source)**

Read-write.

Counts fills into the DC that were initiated by demand ops, per data source.

PMCx043

Bits	Description
7	<b>AlternateMemories_NearFar.</b> Read-write. Requests that return from Extension Memory.
6	<b>DramIO_Far.</b> Read-write. Requests that target another NUMA node and return from DRAM or MMIO.
5	Reserved.
4	<b>NearFarCache_Far.</b> Read-write. Requests that target another NUMA node and return from another CCX's cache.
3	<b>DramIO_Near.</b> Read-write. Requests that target the same NUMA node and return from DRAM or MMIO.
2	<b>NearFarCache_Near.</b> Read-write. Requests that target the same NUMA node and return from another CCX's cache.
1	<b>LocalCcx.</b> Read-write. Data returned from L3 or different L2 in the same CCX.
0	<b>LocalL2.</b> Read-write. Data returned from local L2.

**PMCx044 [Any Data Cache Fills by Data Source] (Core::X86::Pmc::Core::Any\_DC\_Fills\_by\_Data\_Source)**

Read-write.

Counts all fills into the DC, per data source.

PMCx044

Bits	Description
7	<b>AlternateMemories_NearFar</b> . Read-write. Requests that return from Extension Memory.
6	<b>DramIO_Far</b> . Read-write. Requests that target another NUMA node and return from DRAM or MMIO.
5	Reserved.
4	<b>NearFarCache_Far</b> . Read-write. Requests that target another NUMA node and return from another CCX's cache.
3	<b>DramIO_Near</b> . Read-write. Requests that target the same NUMA node and return from DRAM or MMIO.
2	<b>NearFarCache_Near</b> . Read-write. Requests that target the same NUMA node and return from another CCX's cache.
1	<b>LocalCcx</b> . Read-write. Data returned from L3 or different L2 in the same CCX.
0	<b>LocalL2</b> . Read-write. Data returned from local L2.

**PMCx045 [L1 DTLB Reloads] (Core::X86::Pmc::Core::L1\_DTLB\_Reloads)**

Read-write.

Counts L1DTLB reloads

PMCx045

Bits	Description
7	<b>TlbReload1GL2Miss</b> . Read-write. DTLB reload to a 1G page that missed in the L2DTLB.
6	<b>TlbReload2ML2Miss</b> . Read-write. DTLB reload to a 2M page that missed in the L2DTLB.
5	<b>TlbReloadCoalescedPageMiss</b> . Read-write. DTLB reload to a coalesced page that missed in the L2DTLB.
4	<b>TlbReload4KL2Miss</b> . Read-write. DTLB reload to a 4K page that missed in the L2DTLB.
3	<b>TlbReload1GL2Hit</b> . Read-write. DTLB reload to a 1G page that hit in the L2DTLB.
2	<b>TlbReload2ML2Hit</b> . Read-write. DTLB reload to a 2M page that hit in the L2DTLB.
1	<b>TlbReloadCoalescedPageHit</b> . Read-write. DTLB reload to a coalesced page that hit in the L2DTLB.
0	<b>TlbReload4KL2Hit</b> . Read-write. DTLB reload to a 4K page that hit in the L2DTLB.

**PMCx047 [Misaligned Load Flows] (Core::X86::Pmc::Core::Misaligned\_Load\_Flows)**

Read-write.

The number of misaligned load flows.

PMCx047

Bits	Description
7:2	Reserved.
1	<b>MA4K</b> . Read-write. The number of 4KB misaligned (i.e., page crossing) loads or LdOpSt.
0	<b>MA64</b> . Read-write. The number of 64B misaligned (i.e., cacheline crossing) loads or LdOpSt.

**PMCx04B [Prefetch Instructions Dispatched] (Core::X86::Pmc::Core::Software\_Prefetch\_Dispatched)**

Read-write.

Software Prefetch Instructions Dispatched (speculative)

PMCx04B

Bits	Description
7:3	Reserved.
2	<b>PREFETCHNTA</b> . Read-write. PrefetchNTA instruction. See docAPM3 PREFETCHlevel.
1	<b>PREFETCHW</b> . Read-write. PrefetchW instruction. See docAPM3 PREFETCHlevel.
0	<b>PREFETCH</b> . Read-write. PrefetchT0, T1, and T2 instructions. See docAPM3 PREFETCHlevel.

**PMCx050 [Write Combining Buffer Close] (Core::X86::Pmc::Core::WCB\_Close)**

Read-write.

Counts events that cause a Write Combining Buffer (WCB) entry to close.

PMCx050

Bits	Description
7:1	Reserved.
0	<b>FullLine64B.</b> Read-write. All 64 bytes of the WCB entry have been written.

**PMCx052 [Ineffective Software Prefetches] (Core::X86::Pmc::Core::Ineffective\_Software\_Prefetches)**

Read-write.

The number of software prefetches that did not fetch data outside of the processor core.

PMCx052

Bits	Description
7:2	Reserved.
1	<b>MabHit.</b> Read-write. Software PREFETCH instruction saw a match on an already-allocated miss request.
0	<b>DcHit.</b> Read-write. Software PREFETCH instruction saw a DC hit.

**PMCx059 [Software Prefetch Data Cache Fills by Data Source] (Core::X86::Pmc::Core::Software\_Prefetch\_Data\_Cache\_Fills)**

Read-write.

Counts fills into the DC that were initiated by software prefetch instructions, per data source.

PMCx059

Bits	Description
7	<b>AlternateMemories_NearFar.</b> Read-write. Requests that return from Extension Memory.
6	<b>DramIO_Far.</b> Read-write. Requests that target another NUMA node and return from DRAM or MMIO.
5	Reserved.
4	<b>NearFarCache_Far.</b> Read-write. Requests that target another NUMA node and return from another CCX's cache.
3	<b>DramIO_Near.</b> Read-write. Requests that target the same NUMA node and return from DRAM or MMIO.
2	<b>NearFarCache_Near.</b> Read-write. Requests that target the same NUMA node and return from another CCX's cache.
1	<b>LocalCcx.</b> Read-write. Data returned from L3 or different L2 in the same CCX.
0	<b>LocalL2.</b> Read-write. Data returned from local L2.

**PMCx05A [Hardware Prefetch Data Cache Fills by Data Source] (Core::X86::Pmc::Core::Hardware\_Prefetch\_Data\_Cache\_Fills)**

Read-write.

Counts fills into the DC that were initiated by hardware prefetches, per data source.

PMCx05A

Bits	Description
7	<b>AlternateMemories_NearFar.</b> Read-write. Requests that return from Extension Memory.
6	<b>DramIO_Far.</b> Read-write. Requests that target another NUMA node and return from DRAM or MMIO.
5	Reserved.
4	<b>NearFarCache_Far.</b> Read-write. Requests that target another NUMA node and return from another CCX's cache.
3	<b>DramIO_Near.</b> Read-write. Requests that target the same NUMA node and return from DRAM or MMIO.
2	<b>NearFarCache_Near.</b> Read-write. Requests that target the same NUMA node and return from another CCX's cache.
1	<b>LocalCcx.</b> Read-write. Data returned from L3 or different L2 in the same CCX.
0	<b>LocalL2.</b> Read-write. Data returned from local L2.



**PMCx05F [Allocated DC misses] (Core::X86::Pmc::Core::Allocated\_DC\_misses)**

Read-write.

Counts the number of in-flight DC misses each cycle.

PMCx05F

**Bits Description**

7:0 Reserved.

**PMCx076 [Cycles Not in Halt] (Core::X86::Pmc::Core::Cycles\_Not\_in\_Halt)**

Read-write.

Counts cycles when the thread is not in a HALTED state

PMCx076

**Bits Description**

7:0 Reserved.

**PMCx078 [All TLB Flushes] (Core::X86::Pmc::Core::TLB\_Flush\_Events)**

Read-write.

TLB flush events.

PMCx078

**Bits Description**

7:0 All. Read-write. All TLB Flushes

**ValidValues:****Value Description**

FEh-00h Reserved.

FFh Counts all TLB Flushes

**PMCx120 [P0 Freq Cycles not in Halt] (Core::X86::Pmc::Core::P0\_frequency\_Cycles\_Not\_in\_Halt)**

Read-write.

Counts cycles not in Halt, at the P0 P-state frequency, regardless of the current Pstate.

PMCx120

**Bits Description**

7:1 Reserved.

0 **P0\_frequency\_Cycles\_Not\_in\_Halt.** Read-write. Counts at the P0 frequency (same as Core::X86::Msr::MPERF) when not in Halt.**2.1.14.5.3 Instruction Cache (IC) and Branch Prediction (BP) Events**

Note: All instruction cache events are speculative events unless specified otherwise.

**PMCx082 [Instruction Cache Refills From L2] (Core::X86::Pmc::Core::Instruction\_Cache\_Refills\_from\_L2)**

Read-write.

The number of 64 byte instruction cache lines fulfilled from the L2 cache.

PMCx082

**Bits Description**

7:0 Reserved.

**PMCx083 [Instruction Cache Refills from System]  
(Core::X86::Pmc::Core::Instruction\_Cache\_Refills\_from\_System)**

Read-write.

The number of 64 byte instruction cache line fulfilled from system memory or another cache.

PMCx083

Bits	Description
7:0	Reserved.

**PMCx084 [L1 ITLB Miss, L2ITLB Hit] (Core::X86::Pmc::Core::L1\_ITLB\_Miss\_L2\_ITLB\_Hit)**

Read-write.

The number of instruction fetches that miss in the L1 ITLB but hit in the L2 ITLB.

PMCx084

Bits	Description
7:0	Reserved.

**PMCx085 [L1 ITLB Miss, L2 ITLB Miss] (Core::X86::Pmc::Core::ITLB\_Reload\_from\_Page\_Table\_walk)**

Read-write.

The number of instruction fetches that miss in both the L1 ITLB and L2 ITLB.

PMCx085

Bits	Description
7:4	Reserved.
3	<b>Coalesced_4k.</b> Read-write. Walk for >4k Coalesced page (implemented as 16k)
2	<b>walk_1G.</b> Read-write. Walk for 1G page
1	<b>walk_2M.</b> Read-write. Walk for 2M page
0	<b>walk_4K.</b> Read-write. Walk to 4k page

**PMCx08B [BP Pipe Correction or Cancel] (Core::X86::Pmc::Core::BP\_Correct)**

Reset: 00h.

The Branch Predictor flushed its own pipeline due to internal conditions such as a second level prediction structure. Does not count the number of bubbles caused by these internal flushes.

PMCx08B

Bits	Description
7:0	Reserved.

**PMCx08E [Variable Target Predictions] (Core::X86::Pmc::Core::Variable\_Target\_Predictions)**

Read-write.

The number of times a branch used the indirect predictor to make a prediction.

PMCx08E

Bits	Description
7:0	Reserved.

**PMCx091 [Early Redirects]  
(Core::X86::Pmc::Core::Decoder\_Overrides\_Existing\_Branch\_Prediction\_Speculative)**

Reset: 00h.

Number of times that an Early Redirect is sent to Branch Predictor. This happens when either the decoder or dispatch logic is able to detect that the Branch Predictor needs to be redirected.

PMCx091

Bits	Description
7:0	Reserved.

**PMCx094 [ITLB Instruction Fetch Hits] (Core::X86::Pmc::Core::ITLB\_Hits)**

Read-write.

The number of instruction fetches that hit in the L1ITLB.

PMCx094

Bits	Description
7:3	Reserved.
2	<b>IF1G.</b> Read-write. L1 Instruction TLB Hit (1G page size)
1	<b>IF2M.</b> Read-write. L1 Instruction TLB Hit (2M page size)
0	<b>IF4K.</b> Read-write. L1 Instruction TLB Hit (4k or 16k coalesced page size)

**PMCx09F [BP Redirects] (Core::X86::Pmc::Core::BP\_redirects)**

Read-write.

Counts redirects of the branch predictor. To support legacy software, counts both EX mispredict and resyncs when unit\_mask[7:0] is set to 0.

PMCx09F

Bits	Description
7:2	Reserved.
1	<b>ExRedir.</b> Read-write. Mispredict redirect from EX (execution-time)
0	<b>Resync.</b> Read-write. Resync redirect (Retire-time) from RT

**PMCx188 [Fetch IBS events] (Core::X86::Pmc::Core::Fetch\_IBS\_events)**

Read-write.

Counts significant Fetch IBS State transitions.

PMCx188

Bits	Description
7:5	Reserved.
4	<b>SampleVal.</b> Read-write. Counts the number of valid Fetch Instruction Based Sampling (fetch IBS) samples that were collected. Each valid sample also created an IBS interrupt.
3	<b>SampleFiltered.</b> Read-write. Counts the number of Fetch IBS tagged fetches that were discarded due to IBS filtering. When a tagged fetch is discarded the Fetch IBS facility will automatically tag a new fetch.
2	<b>SampleDiscarded.</b> Read-write. Counts when the Fetch IBS facility discards an IBS tagged fetch for reasons other than IBS filtering. When a tagged fetch is discarded the Fetch IBS facility will automatically tag a new fetch.
1	<b>FetchTagged.</b> Read-write. Counts the number of fetches tagged for Fetch IBS. Not all tagged fetches create an IBS interrupt and valid fetch sample.
0	Reserved.

**PMCx18E [IC Tag Hit and Miss Events] (Core::X86::Pmc::Core::IC\_Tag\_Hit\_Miss\_events)**

Read-write.

Counts the number of microtag and full tag events as selected by unit mask.

PMCx18E

Bits	Description
7:5	Reserved.
4:0	<b>IcAccessTypes.</b> Read-write. Instruction Cache accesses.
<b>ValidValues:</b>	
Value	Description
06h-00h	Reserved.
07h	Instruction Cache Hit.
17h-08h	Reserved.
18h	Instruction Cache Miss.
1Eh-19h	Reserved.
1Fh	All Instruction Cache Accesses.

**PMCx28F [Op Cache Hit or Miss] (Core::X86::Pmc::Core::Op\_Cache\_hit\_miss)**

Read-write.

Counts Op Cache micro-tag hit/miss events.

PMCx28F

Bits	Description
7:3	Reserved.
2:0	<b>OpCacheAccesses.</b> Read-write. OpCacheAccesses
<b>ValidValues:</b>	
Value	Description
2h-0h	Reserved.
3h	Op Cache Hit.
4h	Op Cache Miss.
6h-5h	Reserved.
7h	All Op Cache accesses.

**2.1.14.5.4 DE Events****PMCx0A9 [Op Queue Empty] (Core::X86::Pmc::Core::Dispatch\_Empty)**

Reset: 00h.

Cycles where the Op Queue is empty.

PMCx0A9

Bits	Description
7:0	Reserved.

**PMCx0AA [Source of Op Dispatched From Decoder]**  
**(Core::X86::Pmc::Core::Source\_of\_Op\_Dispatched\_From\_Decoder)**

Read-write.

Counts the number of ops dispatched from the decoder classified by op source.

PMCx0AA

Bits	Description
7:2	Reserved.
1	<b>Op_Cache.</b> Read-write. Count of ops dispatched from OpCache
0	<b>x86_decoder.</b> Read-write. Count of ops dispatched from x86 decoder

**PMCx0AB [Types of Ops Dispatched From Decoder]**  
**(Core::X86::Pmc::Core::Types\_of\_Ops\_Dispatched\_From\_Decoder)**

Read-write.

Counts the number of ops dispatched from the decoder classified by op type. The UnitMask value encodes which types of ops are counted.

PMCx0AB

Bits	Description
7:5	Reserved.
4:0	<b>DispOpType.</b> Read-write. DispOpType.
<b>ValidValues:</b>	
<b>Value</b>	<b>Description</b>
03h-00h	Reserved.
04h	Any FP dispatch.
07h-05h	Reserved.
08h	Any Integer dispatch.
1Fh-09h	Reserved.

**PMCx0AE [Dynamic Tokens Dispatch Stall Cycles 1]**  
**(Core::X86::Pmc::Core::Dispatch\_Stall\_Cycles\_Dynamic\_Tokens\_Part\_1)**

Read-write.

Cycles where a dispatch group is valid but does not get dispatched due to a Token Stall. UnitMask bits select the stall types included in the count.

PMCx0AE

Bits	Description
7	Reserved.
6	<b>FPSchRsrcStall.</b> Read-write. FP NSQ token stall
5	Reserved.
4	<b>TakenBrnchBufferRsrc.</b> Read-write. taken branch buffer resource stall.
3	Reserved.
2	<b>StoreQueueRsrcStall.</b> Read-write. STQ Tokens unavailable
1	<b>LoadQueueRsrcStall.</b> Read-write. Load Queue Token Stall.
0	<b>IntPhyRegFileRsrcStall.</b> Read-write. Integer Physical Register File resource stall.

**PMCx0AF [Dynamic Tokens Dispatch Stall Cycles 2] (Core::X86::Pmc::Core::Dispatch\_Stall\_Cycles\_Dynamic\_Tokens\_Part\_2)**

Read-write.

Cycles where a dispatch group is valid but does not get dispatched due to a token stall. UnitMask bits select the stall types included in the count.

PMCx0AF

Bits	Description
7:6	Reserved.
5	<b>RetQ.</b> Read-write. Retire queue tokens unavailable
4:3	Reserved.
2	<b>EX_Flush_recovery.</b> Read-write. Integer Execution flush recovery pending
1	<b>AGTokens.</b> Read-write. Agen tokens unavailable
0	<b>ALTokens.</b> Read-write. ALU tokens unavailable

**PMCx1A0 [No\_Dispatch\_per\_Slot] (Core::X86::Pmc::Core::No\_Dispatch\_per\_Slot)**

Read-write.

Counts the number of dispatch slots (each cycle) that remained unused for reasons selected by UnitMask.

PMCx1A0

Bits	Description																
7:0	<b>StallReason.</b> Read-write. <b>ValidValues:</b>																
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00h</td><td>Reserved.</td></tr> <tr> <td>01h</td><td>Counts dispatch slots left empty because the front-end did not supply ops.</td></tr> <tr> <td>1Dh-02h</td><td>Reserved.</td></tr> <tr> <td>1Eh</td><td>Counts ops unable to dispatch due to back-end stalls.</td></tr> <tr> <td>5Fh-1Fh</td><td>Reserved.</td></tr> <tr> <td>60h</td><td>Counts ops unable to dispatch because the dispatch cycle was granted to the other SMT thread.</td></tr> <tr> <td>FFh-61h</td><td>Reserved.</td></tr> </table>	Value	Description	00h	Reserved.	01h	Counts dispatch slots left empty because the front-end did not supply ops.	1Dh-02h	Reserved.	1Eh	Counts ops unable to dispatch due to back-end stalls.	5Fh-1Fh	Reserved.	60h	Counts ops unable to dispatch because the dispatch cycle was granted to the other SMT thread.	FFh-61h	Reserved.
Value	Description																
00h	Reserved.																
01h	Counts dispatch slots left empty because the front-end did not supply ops.																
1Dh-02h	Reserved.																
1Eh	Counts ops unable to dispatch due to back-end stalls.																
5Fh-1Fh	Reserved.																
60h	Counts ops unable to dispatch because the dispatch cycle was granted to the other SMT thread.																
FFh-61h	Reserved.																

**PMCx1A2 [Dispatch Additional Resource Stalls] (Core::X86::Pmc::Core::Additional\_Resource\_Stalls)**

Read-write.

This PMC event counts additional resource stalls that are not captured by Dispatch\_Stall\_Cycle\_Dynamic\_Tokens\_Part\_1 or Dispatch\_Stall\_Cycles\_Dynamic\_Tokens\_Part\_2.

PMCx1A2

Bits	Description								
7:0	<b>Stall.</b> Read-write. <b>ValidValues:</b>								
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>2Fh-00h</td><td>Reserved.</td></tr> <tr> <td>30h</td><td>Counts additional cycles dispatch is stalled due to the lack of dispatch resources.</td></tr> <tr> <td>FFh-31h</td><td>Reserved.</td></tr> </table>	Value	Description	2Fh-00h	Reserved.	30h	Counts additional cycles dispatch is stalled due to the lack of dispatch resources.	FFh-31h	Reserved.
Value	Description								
2Fh-00h	Reserved.								
30h	Counts additional cycles dispatch is stalled due to the lack of dispatch resources.								
FFh-31h	Reserved.								

**PMCxFFF [Merge] (Core::X86::Pmc::Core::Merge)**

See 2.1.14.4 [Large Increment per Cycle Events].

PMCxFFF

**Bits Description**

7:0 Reserved.

**2.1.14.5.5 EX (SC) Events****PMCx0C0 [Retired Instructions] (Core::X86::Pmc::Core::Retired\_Instructions)**

Read-write.

The number of instructions retired.

PMCx0C0

**Bits Description**

7:0 Reserved.

**PMCx0C1 [Retired Macro-Ops] (Core::X86::Pmc::Core::Retired\_Macro\_Ops)**

Read-write.

The number of macro-ops retired.

PMCx0C1

**Bits Description**

7:0 Reserved.

**PMCx0C2 [Retired Branch Instructions] (Core::X86::Pmc::Core::Retired\_Branch\_Instructions)**

Read-write.

The number of branch instructions retired. This includes all types of architectural control flow changes, including exceptions and interrupts.

PMCx0C2

**Bits Description**

7:0 Reserved.

**PMCx0C3 [Retired Branch Instructions Mispredicted.]  
(Core::X86::Pmc::Core::Retired\_Branch\_Instructions\_Mispredicted)**

Read-write.

The number of retired branch instructions, that were mispredicted. Note that only EX mispredicts are counted.

PMCx0C3

**Bits Description**

7:0 Reserved.

**PMCx0C4 [Retired Taken Branch Instructions] (Core::X86::Pmc::Core::Retired\_Taken\_Branch\_Instructions)**

Read-write.

The number of taken branches that were retired. This includes all types of architectural control flow changes, including exceptions and interrupts.

PMCx0C4

**Bits Description**

7:0 Reserved.

**PMCx0C5 [Retired Taken Branch Instructions Mispredicted.]**  
**(Core::X86::Pmc::Core::Retired\_Taken\_Branch\_Instructions\_Mispredicted)**

Read-write.

The number of retired taken branch instructions that were mispredicted. Note that only EX mispredicts are counted.

PMCx0C5

Bits	Description
------	-------------

7:0	Reserved.
-----	-----------

**PMCx0C6 [Retired Far Control Transfers] (Core::X86::Pmc::Core::Retired\_Far\_Control\_Transfers)**

Read-write.

The number of far control transfers retired including far call/jump/return, IRET, SYSCALL and SYSRET, plus exceptions and interrupts. Far control transfers are not subject to branch prediction.

PMCx0C6

Bits	Description
------	-------------

7:0	Reserved.
-----	-----------

**PMCx0C8 [Retired Near Return Branch Instructions]**  
**(Core::X86::Pmc::Core::Retired\_Near\_Return\_Branch\_Instructions)**

Read-write.

The number of near return instructions (RET [C3] or RET Iw [C2]) retired.

PMCx0C8

Bits	Description
------	-------------

7:0	Reserved.
-----	-----------

**PMCx0C9 [Retired Near Return Branch Instructions Mispredicted]**  
**(Core::X86::Pmc::Core::Retired\_Near\_Return\_Branch\_Instructions\_Mispredicted)**

Read-write.

The number of near returns retired that were not correctly predicted by the return address predictor. Each such mispredict incurs the same penalty as a mispredicted conditional branch instruction. Note that only EX mispredicts are counted .

PMCx0C9

Bits	Description
------	-------------

7:0	Reserved.
-----	-----------

**PMCx0CA [Retired Indirect Branch Instructions Mispredicted]**  
**(Core::X86::Pmc::Core::Retired\_Indirect\_Branch\_Instructions\_Mispredicted)**

Read-write.

The number of indirect branches retired that were not correctly predicted. Each such mispredict incurs the same penalty as a mispredicted conditional branch instruction. Note that only EX mispredicts are counted .

PMCx0CA

Bits	Description
------	-------------

7:0	Reserved.
-----	-----------



**PMCx0CB [Retired MMX FP Instructions] (Core::X86::Pmc::Core::Retired\_MMX\_FP\_Instructions)**

Read-write.

The number of MMX, SSE or x87 instructions retired. The UnitMask allows the selection of the individual classes of instructions as given in the table. Each increment represents one complete instruction. Since this event includes non-numeric instructions it is not suitable for measuring MFLOPs

PMCx0CB

Bits	Description
7:3	Reserved.
2	SSE. Read-write. SSE instructions (SSE, SSE2, SSE3, SSSE3, SSE4A, SSE41, SSE42, AVX).
1	MMX. Read-write. MMX instructions
0	X87. Read-write. x87 instructions

**PMCx0CC [Retired Indirect Branch Instructions]  
(Core::X86::Pmc::Core::Retired\_Indirect\_Branch\_Instructions)**

Read-write.

The number of indirect branches retired.

PMCx0CC

Bits	Description
7:0	Reserved.

**PMCx0D1 [Retired Conditional Branch Instructions]  
(Core::X86::Pmc::Core::Retired\_Conditional\_Branch\_Instructions)**

Read-write.

Count of conditional branch instructions that retired

PMCx0D1

Bits	Description
7:0	Reserved.

**PMCx0D3 [Div Cycles Busy count] (Core::X86::Pmc::Core::Div\_Cycles\_Busy\_count)**

Read-write.

Counts cycles when the divider is busy

PMCx0D3

Bits	Description
7:0	Reserved.

**PMCx0D4 [Div Op Count] (Core::X86::Pmc::Core::Div\_Op\_Count)**

Read-write.

Counts number of divide ops

PMCx0D4

Bits	Description
7:0	Reserved.

**PMCx0D6 [Cycles with no retire] (Core::X86::Pmc::Core::Cycles\_With\_No\_Retire)**

Read-write.

This event counts cycles when the hardware thread does not retire any ops for reasons selected by UnitMask[4:0]. UnitMask events [4:0] are mutually exclusive. If multiple reasons apply for a given cycle, the lowest numbered UnitMask event is counted.

PMCx0D6

Bits	Description										
7:5	<b>CompletionFilter.</b> Read-write. <b>ValidValues:</b>										
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>Load and ALU completion is considered for UnitMask[1]: NotComplete events.</td></tr> <tr> <td>4h-1h</td><td>Reserved.</td></tr> <tr> <td>5h</td><td>Only missing Load completion is considered for UnitMask[1]: NotComplete events.</td></tr> <tr> <td>7h-6h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	Load and ALU completion is considered for UnitMask[1]: NotComplete events.	4h-1h	Reserved.	5h	Only missing Load completion is considered for UnitMask[1]: NotComplete events.	7h-6h	Reserved.
Value	Description										
0h	Load and ALU completion is considered for UnitMask[1]: NotComplete events.										
4h-1h	Reserved.										
5h	Only missing Load completion is considered for UnitMask[1]: NotComplete events.										
7h-6h	Reserved.										
4	<b>ThreadNotSelected.</b> Read-write. The number cycles where ops could have retired (i.e. did not fall into the sub-events [0]...[3]) but did not retire because the thread arbitration did not select the thread for retire.										
3	<b>Other.</b> Read-write. The number of cycles where ops could have retired (self and older ops are complete), but were stopped from retirement for other reasons: retire breaks, traps, faults, etc.										
2	Reserved.										
1	<b>NotCompleteSelf.</b> Read-write. The number of cycles where the oldest retire slot did not have its completion bits set.										
0	<b>Empty.</b> Read-write. The number of cycles when there were no valid ops in the retire queue. This may be caused by front-end bottlenecks or pipeline redirects.										

**PMCx1C1 [Retired Microcoded Instructions] (Core::X86::Pmc::Core::Retired\_Microcoded\_Instructions)**

Read-write.

The number of retired microcoded instructions.

PMCx1C1

Bits	Description
7:0	Reserved.

**PMCx1C2 [Retired Microcode Ops] (Core::X86::Pmc::Core::Retired\_Microcode\_Ops)**

Read-write.

The number of microcode ops that have retired.

PMCx1C2

Bits	Description
7:0	Reserved.

**PMCx1C7 [Retired Conditional Branch Instructions Mispredicted] (Core::X86::Pmc::Core::Retired\_Conditional\_Branch\_Instructions\_Mispredicted)**

Read-write.

The number of retired conditional branch instructions that were not correctly predicted because of a branch direction mismatch.

PMCx1C7

Bits	Description
7:0	Reserved.

**PMCx1C8 [Retired Unconditional Branch Instructions Mispredicted]  
(Core::X86::Pmc::Core::Retired\_Unconditional\_Branch\_Instructions\_Mispredicted)**

Read-write.

The number of retired unconditional indirect branch instructions that were mispredicted.

PMCx1C8

Bits	Description
------	-------------

7:0	Reserved.
-----	-----------

**PMCx1C9 [Retired Unconditional Branch Instructions]  
(Core::X86::Pmc::Core::Retired\_Unconditional\_Branch\_Instructions)**

Read-write.

PMCx1C9

Bits	Description
------	-------------

7:0	Reserved.
-----	-----------

**PMCx1CF [Tagged IBS Ops] (Core::X86::Pmc::Core::Tagged\_IBS\_Ops)**

Read-write.

Counts Op IBS related events

PMCx1CF

Bits	Description
------	-------------

7:3	Reserved.
-----	-----------

2	<b>IbsCountRollover.</b> Read-write. Number of times an op could not be tagged by IBS because of a previous tagged op that has not yet signaled interrupt.
---	--

1	<b>IbsTaggedOpsRet.</b> Read-write. Number of Ops tagged by IBS that retired
---	--

0	<b>IbsTaggedOps.</b> Read-write. Number of Ops tagged by IBS
---	--

**PMCx1D0 [Retired Fused Instructions] (Core::X86::Pmc::Core::Retired\_fused\_instructions)**

Reset: 00h.

Counts retired fused instructions.

PMCx1D0

Bits	Description
------	-------------

7:0	Reserved.
-----	-----------

## 2.1.14.5.6 L2 Cache Events

**PMCx060 [Requests to L2 Group1] (Core::X86::Pmc::L2::L2RequestG1)**

Read-write.

All L2 Cache Requests (Breakdown 1 - Common)

PMCx060

Bits	Description
------	-------------

7	<b>RdBlkL.</b> Read-write. Data Cache Reads (including hardware and software prefetch).
---	---

6	<b>RdBlkX.</b> Read-write. Data Cache Stores
---	--

5	<b>LsRdBlkC_S.</b> Read-write. Data Cache Shared Reads
---	--

4	<b>CacheableIcRead.</b> Read-write. Instruction Cache Reads.
---	--

3	Reserved.
---	-----------

2	<b>LsPrefetchL2Cmd.</b> Read-write.
---	-------------------------------------

1	<b>L2HwPf: L2 Prefetcher.</b> Read-write. All prefetches accepted by L2 pipeline, hit or miss. Types of PF and L2 hit/miss broken out in a separate perfmon event
---	---

0	<b>Group2.</b> Read-write. MiscRequests. Read-write. Various Noncacheable requests. Non-cached Data Reads, Non-cached Instruction Reads, Self-modifying code checks.
---	--

**PMCx061 [Requests to L2 Group2] (Core::X86::Pmc::L2::L2RequestG2)**

Read-write.

All L2 Cache Requests (Breakdown 2 - Rare).

PMCx061

Bits	Description
7	Reserved.
6	<b>LsRdSized</b> . Read-write. LS sized read, coherent non-cacheable.
5	<b>LsRdSizedNC</b> . Read-write. LS sized read, non-coherent, non-cacheable.
4:0	Reserved.

**PMCx063 [Write Combining Buffer Requests] (Core::X86::Pmc::L2::L2WcbReq)**

Read-write.

Write Combining Buffer operations. For information on Write Combining see docAPM2 sections: Memory System, Memory Types, Buffering and Combining Memory Writes.

PMCx063

Bits	Description
7:6	Reserved.
5	<b>WcbClose</b> . Read-write. Write Combining Buffer close
4:0	Reserved.

**PMCx064 [Core to L2 Cacheable Request Access Status] (Core::X86::Pmc::L2::L2CacheReqStat)**

Read-write.

L2 Cache Request Outcomes (not including L2 Prefetch).

PMCx064

Bits	Description
7	<b>LsRdBlkCS: Data Cache Shared Read Hit in L2</b> . Read-write. LsRdBlkCS
6	<b>LsRdBlkLHitX: Data Cache Read Hit in L2</b> . Read-write. Modifiable
5	<b>LsRdBlkLHitS: Data Cache Read Hit Non-Modifiable Line in L2</b> . Read-write.
4	<b>LsRdBlkX: Data Cache Store Hit in L2</b> . Read-write.
3	<b>LsRdBlkC: Data Cache Req Miss in L2</b> . Read-write.
2	<b>IcFillHitX: Instruction Cache Hit Modifiable Line in L2</b> . Read-write. IcFillHitX
1	<b>IcFillHitS: Instruction Cache Hit Non-Modifiable Line in L2</b> . Read-write.
0	<b>IcFillMiss: Instruction Cache Req Miss in L2</b> . Read-write. IcFillMiss

**PMCx070 [L2 Prefetch Hit in L2] (Core::X86::Pmc::L2::L2PfHitL2)**

Read-write.

Counts all L2 prefetches accepted by L2 pipeline which hit in the L2 cache.

PMCx070

Bits	Description														
7:0	<b>Prefetches</b> . Read-write.														
<b>ValidValues:</b>															
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>1Eh-00h</td><td>Reserved.</td></tr> <tr> <td>1Fh</td><td>Counts requests generated from L2 Hardware Prefetchers.</td></tr> <tr> <td>DFh-20h</td><td>Reserved.</td></tr> <tr> <td>E0h</td><td>Counts requests generated from L1 DC Hardware Prefetchers.</td></tr> <tr> <td>FEh-E1h</td><td>Reserved.</td></tr> <tr> <td>FFh</td><td>Counts requests generated from L1 DC and L2 Hardware Prefetchers.</td></tr> </table>	Value	Description	1Eh-00h	Reserved.	1Fh	Counts requests generated from L2 Hardware Prefetchers.	DFh-20h	Reserved.	E0h	Counts requests generated from L1 DC Hardware Prefetchers.	FEh-E1h	Reserved.	FFh	Counts requests generated from L1 DC and L2 Hardware Prefetchers.
Value	Description														
1Eh-00h	Reserved.														
1Fh	Counts requests generated from L2 Hardware Prefetchers.														
DFh-20h	Reserved.														
E0h	Counts requests generated from L1 DC Hardware Prefetchers.														
FEh-E1h	Reserved.														
FFh	Counts requests generated from L1 DC and L2 Hardware Prefetchers.														

**PMCx071 [L2 Prefetcher Hits in L3] (Core::X86::Pmc::L2::L2PfMissL2HitL3)**

Read-write.

Counts all L2 prefetches accepted by the L2 pipeline which miss the L2 cache and hit the L3.

PMCx071

Bits	Description
7:0	<b>Prefetches.</b> Read-write. L2Stream
<b>ValidValues:</b>	
Value	Description
1Eh-00h	Reserved.
1Fh	Counts requests generated from L2 Hardware Prefetchers.
DFh-20h	Reserved.
E0h	Counts requests generated from L1 DC Hardware Prefetchers.
FEh-E1h	Reserved.
FFh	Counts requests generated from L1 DC and L2 Hardware Prefetchers.

**PMCx072 [L2 Prefetcher Misses in L3] (Core::X86::Pmc::L2::L2PfMissL2L3)**

Read-write.

Counts all L2 prefetches accepted by the L2 pipeline which miss the L2 and the L3 caches

PMCx072

Bits	Description
7:0	<b>Prefetches.</b> Read-write. L2Stream
<b>ValidValues:</b>	
Value	Description
1Eh-00h	Reserved.
1Fh	Counts requests generated from L2 Hardware Prefetchers.
DFh-20h	Reserved.
E0h	Counts requests generated from L1 DC Hardware Prefetchers.
FEh-E1h	Reserved.
FFh	Counts requests generated from L1 DC and L2 Hardware Prefetchers.

**PMCx165 [L2 Fill Response Source] (Core::X86::Pmc::L2::L2FillRspSrc)**

Read-write.

Counts fill responses based on their source. Selecting an event mask of 0xfe will count all L3 responses.

This will count all L3 responses to fill requests.

This event is similar to LS PMC 0x44

PMCx165

Bits	Description
7	<b>AlternateMemories_NearFar</b> . Read-write. "Requests that return from Extension Memory"
6	<b>DramIO_Far</b> . Read-write. Requests that target another NUMA node and return from either DRAM or MMIO from another NUMA node, either from the same or different NUMA node.
5	Reserved.
4	<b>NearFarCache_Far</b> . Read-write. Requests that target another NUMA node and return from another CCX's cache.
3	<b>DramIO_Near</b> . Read-write. Requests that target the same NUMA node and return from either DRAM or MMIO from the same NUMA node.
2	<b>NearFarCache_Near</b> . Read-write. Requests that target the same NUMA node and return from another CCX's cache.
1	<b>LocalCcx</b> . Read-write. Data returned from L3 or different L2 in the same CCX.
0	Reserved.

**2.1.14.6 L3 Cache Performance Monitor Counters**

This section provides the core performance counter events that may be selected through Core::X86::Msr::ChL3PmcCfg.

- When in non-SMT mode, thread 0 must be selected for events that don't ignore ThreadMask.

**2.1.14.6.1 L3 Cache PMC Events****L3PMCx04 [L3 tag lookup state] (Core::X86::Pmc::L3::L3LookupState)**

Read-write.

All L3 Requests.

L3PMCx04

Bits	Description												
7:0	<b>L3LookupMask</b> . Read-write. L3 Request Types												
<b>ValidValues:</b>													
	<table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>00h</td><td>Reserved.</td></tr> <tr> <td>01h</td><td>L3 Miss</td></tr> <tr> <td>FDh-02h</td><td>Reserved.</td></tr> <tr> <td>FEh</td><td>L3 Hit</td></tr> <tr> <td>FFh</td><td>All coherent accesses to L3</td></tr> </table>	Value	Description	00h	Reserved.	01h	L3 Miss	FDh-02h	Reserved.	FEh	L3 Hit	FFh	All coherent accesses to L3
Value	Description												
00h	Reserved.												
01h	L3 Miss												
FDh-02h	Reserved.												
FEh	L3 Hit												
FFh	All coherent accesses to L3												

**L3PMCxAC [L3\_XiSampledLatency] (Core::X86::Pmc::L3::L3\_XiSampledLatency)**

Read-write.

When used in conjunction with L3\_XiSampledLatencyRequests, this PMC Event will measure the average memory latency (excluding MMIO) observed by this CCX.

Configure two PMCs with the L3\_XiSampledLatency and L3\_XiSampledLatencyRequests events and use the following equation to identify the observed latency.

$$\text{Average Sampled Latency} = \text{L3\_XiSampledLatency} / \text{L3\_XiSampledLatencyRequests} * 10\text{ns}$$

Some ChL3PmcCfg fields must be programmed as follows to ensure that these events accurately measure latency: ChL3PmcCfg[EnAllSources]=0x1.

Other ChL3PmcCfg fields can be used to filter the measured latency based on originating thread (EnAllCores, CoreID) and Data Source (UnitMask).

To measure average latency from all threads to all Data Sources, use the following configuration:

ChL3PmcCfg[EnAllCores]=0x1, ChL3PmcCfg[ThreadMask]=0x3, and ChL3PmcCfg[UnitMask]=0xFF.

L3PMCxAC

Bits	Description
7:6	Reserved.
5	<b>Ext_Far.</b> Read-write. Requests that target another NUMA node and return from Extension Memory (CXL™)
4	<b>Ext_Near.</b> Read-write. Requests that target the same NUMA node and return from Extension Memory (CXL)
3	<b>NearCache_FarCache_Far.</b> Read-write. Requests that target another NUMA node and return from another CCX's cache.
2	<b>NearCache_FarCache_Near.</b> Read-write. Requests that target the same NUMA node and return from another CCX's cache.
1	<b>Dram_Far.</b> Read-write. Requests that target another NUMA node and return from DRAM
0	<b>Dram_Near.</b> Read-write. Requests that target the same NUMA node and return from DRAM

**L3PMCxAD [L3 XiSampledLatencyRequests] (Core::X86::Pmc::L3::L3\_XiSampledLatencyRequests)**

Read-write.

When used in conjunction with L3\_XiSampledLatency, this PMC Event will measure the average memory latency (excluding MMIO) observed by this CCX.

Configure two PMCs with the L3\_XiSampledLatency and L3\_XiSampledLatencyRequests events and use the following equation to identify the observed latency.

$$\text{Average Sampled Latency} = \text{L3\_XiSampledLatency} / \text{L3\_XiSampledLatencyRequests} * 10\text{ns}$$

Some ChL3PmcCfg fields must be programmed as follows to ensure that these events accurately measure latency: ChL3PmcCfg[EnAllSources]=0x1.

Other ChL3PmcCfg fields can be used to filter the measured latency based on originating thread (EnAllCores, CoreID) and Data Source (UnitMask).

To measure average latency from all threads to all Data Sources, use the following configuration:

ChL3PmcCfg[EnAllCores]=0x1, ChL3PmcCfg[ThreadMask]=0x3, and ChL3PmcCfg[UnitMask]=0xFF.

L3PMCxAD

Bits	Description
7:6	Reserved.
5	<b>Ext_Far.</b> Read-write. Requests that target another NUMA node and return from Extension Memory (CXL)
4	<b>Ext_Near.</b> Read-write. Requests that target the same NUMA node and return from Extension Memory (CXL)
3	<b>NearCache_FarCache_Far.</b> Read-write. Requests that target another NUMA node and return from another CCX's cache.
2	<b>NearCache_FarCache_Near.</b> Read-write. Requests that target the same NUMA node and return from another CCX's cache.
1	<b>Dram_Far.</b> Read-write. Requests that target another NUMA node and return from DRAM
0	<b>Dram_Near.</b> Read-write. Requests that target the same NUMA node and return from DRAM

### 2.1.15 Instruction Based Sampling (IBS)

IBS is a code profiling mechanism that enables the processor to select a random instruction fetch or macro-op after a programmed time interval has expired and record specific performance information about the operation. An interrupt is generated when the operation is complete as specified by Core::X86::Msrr::IBS\_CTL. An interrupt handler can then read the performance information that was logged for the operation.

The IBS mechanism is split into two parts: instruction fetch performance controlled by Core::X86::Msrr::IBS\_FETCH\_CTL; and instruction execution performance controlled by Core::X86::Msrr::IBS\_OP\_CTL. Instruction fetch sampling provides information about instruction TLB and instruction cache behavior for fetched instructions. Instruction execution sampling provides information about op execution behavior. The data collected for instruction fetch performance is independent from the data collected for instruction execution performance. Support for the IBS feature is indicated by the Core::X86::Cpuid::FeatureExtIdEcX[IBS].

Instruction fetch performance is profiled by recording the following performance information for the tagged instruction fetch:

- If the instruction fetch completed or was aborted. See Core::X86::Msrr::IBS\_FETCH\_CTL.
- The number of clock cycles spent on the instruction fetch. See Core::X86::Msrr::IBS\_FETCH\_CTL.
- If the instruction fetch hit or missed the IC, hit/missed in the L1 and L2 TLBs, and page size. See Core::X86::Msrr::IBS\_FETCH\_CTL.
- The linear address, physical address associated with the fetch. See Core::X86::Msrr::IBS\_FETCH\_LINADDR, Core::X86::Msrr::IBS\_FETCH\_PHYSADDR.



Instruction execution performance is profiled by tagging one macro-op associated with an instruction. Instructions that decode to more than one macro-op return different performance data depending upon which macro-op associated with the instruction is tagged. These macro-ops are associated with the RIP of the next instruction to retire. The following performance information is returned for the tagged op:

- Branch and execution status. See Core::X86::Msr::IBS\_OP\_DATA.
- Branch target address for branch ops. See Core::X86::Msr::BP\_IBSTGT\_RIP.
- The logical address associated with the op. See Core::X86::Msr::IBS\_OP\_RIP.
- The linear and physical address associated with a load or store op. See Core::X86::Msr::IBS\_DC\_LINADDR, Core::X86::Msr::IBS\_DC\_PHYSADDR.
- The data cache access status associated with the op: DC hit/miss, DC miss latency, TLB hit/miss, TLB page size. See Core::X86::Msr::IBS\_OP\_DATA3.
- The number clocks from when the op was tagged until the op retires. See Core::X86::Msr::IBS\_OP\_DATA.
- The number clocks from when the op completes execution until the op retires. See Core::X86::Msr::IBS\_OP\_DATA.
- Source information for DRAM and MMIO. See Core::X86::Msr::IBS\_OP\_DATA2.

## 2.2 L3 Cache

The Level-3 cache (L3) forms the third level of cache in the CPU caching hierarchy. The L3 is a shared, unified cache inside a core complex.

### 2.2.1 L3 MSR Registers

MSR0000_0C81 [L3 QoS Configuration] (L3::L3CRB::L3QosCfg1)	
Reset: 0000_0000_0000_0000h.	
QOS L3 Cache Allocation CDP mode enable (I vs. D). Contents are copied to ChL2QosCfg1 and ChL3QosCfg1_0.	
_lthree[1:0]; MSR0000_0C81	
Bits	Description
63:1	Reserved.
0	<b>CDP.</b> Read-write. Reset: 0. Code and Data Prioritization Technology enable
MSR0000_0C90 [L3 QoS Allocation Mask] (L3::L3CRB::L3QosAllocMask0)	
Reset: 0000_0000_0000_FFFFh.	
QOS L3 Allocation Mask for CLOS0	
_lthree[1:0]; MSR0000_0C90	
Bits	Description
63:16	Reserved.
15:0	<b>WayMask.</b> Read-write. Reset: FFFFh. L3 way mask used for allocation control.
MSR0000_0C91 [L3 QoS Allocation Mask] (L3::L3CRB::L3QosAllocMask1)	
Reset: 0000_0000_0000_FFFFh.	
QOS L3 Allocation Mask for CLOS1	
_lthree[1:0]; MSR0000_0C91	
Bits	Description
63:16	Reserved.
15:0	<b>WayMask.</b> Read-write. Reset: FFFFh. L3 way mask used for allocation control.

**MSR0000\_0C92 [L3 QoS Allocation Mask] (L3::L3CRB::L3QosAllocMask2)**

Reset: 0000\_0000\_0000\_FFFFh.

QOS L3 Allocation Mask for CLOS2

\_lthree[1:0]; MSR0000\_0C92

**Bits Description**

63:16 Reserved.

15:0 **WayMask.** Read-write. Reset: FFFFh. L3 way mask used for allocation control.**MSR0000\_0C93 [L3 QoS Allocation Mask] (L3::L3CRB::L3QosAllocMask3)**

Reset: 0000\_0000\_0000\_FFFFh.

QOS L3 Allocation Mask for CLOS3

\_lthree[1:0]; MSR0000\_0C93

**Bits Description**

63:16 Reserved.

15:0 **WayMask.** Read-write. Reset: FFFFh. L3 way mask used for allocation control.**MSR0000\_0C94 [L3 QoS Allocation Mask] (L3::L3CRB::L3QosAllocMask4)**

Reset: 0000\_0000\_0000\_FFFFh.

QOS L3 Allocation Mask for CLOS4

\_lthree[1:0]; MSR0000\_0C94

**Bits Description**

63:16 Reserved.

15:0 **WayMask.** Read-write. Reset: FFFFh. L3 way mask used for allocation control.**MSR0000\_0C95 [L3 QoS Allocation Mask] (L3::L3CRB::L3QosAllocMask5)**

Reset: 0000\_0000\_0000\_FFFFh.

QOS L3 Allocation Mask for CLOS5

\_lthree[1:0]; MSR0000\_0C95

**Bits Description**

63:16 Reserved.

15:0 **WayMask.** Read-write. Reset: FFFFh. L3 way mask used for allocation control.**MSR0000\_0C96 [L3 QoS Allocation Mask] (L3::L3CRB::L3QosAllocMask6)**

Reset: 0000\_0000\_0000\_FFFFh.

QOS L3 Allocation Mask for CLOS6

\_lthree[1:0]; MSR0000\_0C96

**Bits Description**

63:16 Reserved.

15:0 **WayMask.** Read-write. Reset: FFFFh. L3 way mask used for allocation control.**MSR0000\_0C97 [L3 QoS Allocation Mask] (L3::L3CRB::L3QosAllocMask7)**

Reset: 0000\_0000\_0000\_FFFFh.

QOS L3 Allocation Mask for CLOS7

\_lthree[1:0]; MSR0000\_0C97

**Bits Description**

63:16 Reserved.

15:0 **WayMask.** Read-write. Reset: FFFFh. L3 way mask used for allocation control.

**MSR0000\_0C98 [L3 QoS Allocation Mask] (L3::L3CRB::L3QosAllocMask8)**

Reset: 0000\_0000\_0000\_FFFFh.

QOS L3 Allocation Mask for CLOS8

\_lthree[1:0]; MSR0000\_0C98

**Bits Description**

63:16 Reserved.

15:0 **WayMask.** Read-write. Reset: FFFFh. L3 way mask used for allocation control.**MSR0000\_0C99 [L3 QoS Allocation Mask] (L3::L3CRB::L3QosAllocMask9)**

Reset: 0000\_0000\_0000\_FFFFh.

QOS L3 Allocation Mask for CLOS9

\_lthree[1:0]; MSR0000\_0C99

**Bits Description**

63:16 Reserved.

15:0 **WayMask.** Read-write. Reset: FFFFh. L3 way mask used for allocation control.**MSR0000\_0C9A [L3 QoS Allocation Mask] (L3::L3CRB::L3QosAllocMask10)**

Reset: 0000\_0000\_0000\_FFFFh.

QOS L3 Allocation Mask for CLOS10

\_lthree[1:0]; MSR0000\_0C9A

**Bits Description**

63:16 Reserved.

15:0 **WayMask.** Read-write. Reset: FFFFh. L3 way mask used for allocation control.**MSR0000\_0C9B [L3 QoS Allocation Mask] (L3::L3CRB::L3QosAllocMask11)**

Reset: 0000\_0000\_0000\_FFFFh.

QOS L3 Allocation Mask for CLOS11

\_lthree[1:0]; MSR0000\_0C9B

**Bits Description**

63:16 Reserved.

15:0 **WayMask.** Read-write. Reset: FFFFh. L3 way mask used for allocation control.**MSR0000\_0C9C [L3 QoS Allocation Mask] (L3::L3CRB::L3QosAllocMask12)**

Reset: 0000\_0000\_0000\_FFFFh.

QOS L3 Allocation Mask for CLOS12

\_lthree[1:0]; MSR0000\_0C9C

**Bits Description**

63:16 Reserved.

15:0 **WayMask.** Read-write. Reset: FFFFh. L3 way mask used for allocation control.**MSR0000\_0C9D [L3 QoS Allocation Mask] (L3::L3CRB::L3QosAllocMask13)**

Reset: 0000\_0000\_0000\_FFFFh.

QOS L3 Allocation Mask for CLOS13

\_lthree[1:0]; MSR0000\_0C9D

**Bits Description**

63:16 Reserved.

15:0 **WayMask.** Read-write. Reset: FFFFh. L3 way mask used for allocation control.

**MSR0000\_0C9E [L3 QoS Allocation Mask] (L3::L3CRB::L3QosAllocMask14)**

Reset: 0000\_0000\_0000\_FFFFh.

QOS L3 Allocation Mask for CLOS14

\_lthree[1:0]; MSR0000\_0C9E

**Bits Description**

63:16 Reserved.

15:0 **WayMask.** Read-write. Reset: FFFFh. L3 way mask used for allocation control.**MSR0000\_0C9F [L3 QoS Allocation Mask] (L3::L3CRB::L3QosAllocMask15)**

Reset: 0000\_0000\_0000\_FFFFh.

QOS L3 Allocation Mask for CLOS15

\_lthree[1:0]; MSR0000\_0C9F

**Bits Description**

63:16 Reserved.

15:0 **WayMask.** Read-write. Reset: FFFFh. L3 way mask used for allocation control.**MSRC000\_0200 [L3 QoS Bandwidth Control] (L3::L3CRB::L3QosBwControl0)**

Reset: 0000\_0000\_0000\_1000h.

QOS BW Control0

\_lthree[1:0]; MSRC000\_0200

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_0201 [L3 QoS Bandwidth Control] (L3::L3CRB::L3QosBwControl1)**

Reset: 0000\_0000\_0000\_1000h.

QOS BW Control1

\_lthree[1:0]; MSRC000\_0201

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_0202 [L3 QoS Bandwidth Control] (L3::L3CRB::L3QosBwControl2)**

Reset: 0000\_0000\_0000\_1000h.

QOS BW Control2

\_lthree[1:0]; MSRC000\_0202

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_0203 [L3 QoS Bandwidth Control] (L3::L3CRB::L3QosBwControl3)**

Reset: 0000\_0000\_0000\_1000h.

QOS BW Control3

\_lthree[1:0]; MSRC000\_0203

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value

**MSRC000\_0204 [L3 QoS Bandwidth Control] (L3::L3CRB::L3QosBwControl4)**

Reset: 0000\_0000\_0000\_1000h.

QOS BW Control4

\_lthree[1:0]; MSRC000\_0204

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_0205 [L3 QoS Bandwidth Control] (L3::L3CRB::L3QosBwControl5)**

Reset: 0000\_0000\_0000\_1000h.

QOS BW Control5

\_lthree[1:0]; MSRC000\_0205

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_0206 [L3 QoS Bandwidth Control] (L3::L3CRB::L3QosBwControl6)**

Reset: 0000\_0000\_0000\_1000h.

QOS BW Control6

\_lthree[1:0]; MSRC000\_0206

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_0207 [L3 QoS Bandwidth Control] (L3::L3CRB::L3QosBwControl7)**

Reset: 0000\_0000\_0000\_1000h.

QOS BW Control7

\_lthree[1:0]; MSRC000\_0207

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_0208 [L3 QoS Bandwidth Control] (L3::L3CRB::L3QosBwControl8)**

Reset: 0000\_0000\_0000\_1000h.

QOS BW Control8

\_lthree[1:0]; MSRC000\_0208

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_0209 [L3 QoS Bandwidth Control] (L3::L3CRB::L3QosBwControl9)**

Reset: 0000\_0000\_0000\_1000h.

QOS BW Control9

\_lthree[1:0]; MSRC000\_0209

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value

**MSRC000\_020A [L3 QoS Bandwidth Control] (L3::L3CRB::L3QosBwControl10)**

Reset: 0000\_0000\_0000\_1000h.

QOS BW Control10

\_lthree[1:0]; MSRC000\_020A

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_020B [L3 QoS Bandwidth Control] (L3::L3CRB::L3QosBwControl11)**

Reset: 0000\_0000\_0000\_1000h.

QOS BW Control11

\_lthree[1:0]; MSRC000\_020B

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_020C [L3 QoS Bandwidth Control] (L3::L3CRB::L3QosBwControl12)**

Reset: 0000\_0000\_0000\_1000h.

QOS BW Control12

\_lthree[1:0]; MSRC000\_020C

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_020D [L3 QoS Bandwidth Control] (L3::L3CRB::L3QosBwControl13)**

Reset: 0000\_0000\_0000\_1000h.

QOS BW Control13

\_lthree[1:0]; MSRC000\_020D

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_020E [L3 QoS Bandwidth Control] (L3::L3CRB::L3QosBwControl14)**

Reset: 0000\_0000\_0000\_1000h.

QOS BW Control14

\_lthree[1:0]; MSRC000\_020E

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_020F [L3 QoS Bandwidth Control] (L3::L3CRB::L3QosBwControl15)**

Reset: 0000\_0000\_0000\_1000h.

QOS BW Control15

\_lthree[1:0]; MSRC000\_020F

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value

**MSRC000\_0280 (L3::L3CRB::L3QOS\_SLOWBW\_CONTROL\_0)**

Reset: 0000\_0000\_0000\_1000h.

QOS Slow Memory BW Control0

\_lthree[1:0]; MSRC000\_0280

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_0281 (L3::L3CRB::L3QOS\_SLOWBW\_CONTROL\_1)**

Reset: 0000\_0000\_0000\_1000h.

QOS Slow Memory BW Control1

\_lthree[1:0]; MSRC000\_0281

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_0282 (L3::L3CRB::L3QOS\_SLOWBW\_CONTROL\_2)**

Reset: 0000\_0000\_0000\_1000h.

QOS Slow Memory BW Control2

\_lthree[1:0]; MSRC000\_0282

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_0283 (L3::L3CRB::L3QOS\_SLOWBW\_CONTROL\_3)**

Reset: 0000\_0000\_0000\_1000h.

QOS Slow Memory BW Control3

\_lthree[1:0]; MSRC000\_0283

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_0284 (L3::L3CRB::L3QOS\_SLOWBW\_CONTROL\_4)**

Reset: 0000\_0000\_0000\_1000h.

QOS Slow Memory BW Control4

\_lthree[1:0]; MSRC000\_0284

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_0285 (L3::L3CRB::L3QOS\_SLOWBW\_CONTROL\_5)**

Reset: 0000\_0000\_0000\_1000h.

QOS Slow Memory BW Control5

\_lthree[1:0]; MSRC000\_0285

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value

**MSRC000\_0286 (L3::L3CRB::L3QOS\_SLOWBW\_CONTROL\_6)**

Reset: 0000\_0000\_0000\_1000h.

QOS Slow Memory BW Control6

\_lthree[1:0]; MSRC000\_0286

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_0287 (L3::L3CRB::L3QOS\_SLOWBW\_CONTROL\_7)**

Reset: 0000\_0000\_0000\_1000h.

QOS Slow Memory BW Control7

\_lthree[1:0]; MSRC000\_0287

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_0288 (L3::L3CRB::L3QOS\_SLOWBW\_CONTROL\_8)**

Reset: 0000\_0000\_0000\_1000h.

QOS Slow Memory BW Control8

\_lthree[1:0]; MSRC000\_0288

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_0289 (L3::L3CRB::L3QOS\_SLOWBW\_CONTROL\_9)**

Reset: 0000\_0000\_0000\_1000h.

QOS Slow Memory BW Control9

\_lthree[1:0]; MSRC000\_0289

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_028A (L3::L3CRB::L3QOS\_SLOWBW\_CONTROL\_10)**

Reset: 0000\_0000\_0000\_1000h.

QOS Slow Memory BW Control10

\_lthree[1:0]; MSRC000\_028A

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_028B (L3::L3CRB::L3QOS\_SLOWBW\_CONTROL\_11)**

Reset: 0000\_0000\_0000\_1000h.

QOS Slow Memory BW Control11

\_lthree[1:0]; MSRC000\_028B

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value



**MSRC000\_028C (L3::L3CRB::L3QOS\_SLOWBW\_CONTROL\_12)**

Reset: 0000\_0000\_0000\_1000h.

QOS Slow Memory BW Control12

\_lthree[1:0]; MSRC000\_028C

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_028D (L3::L3CRB::L3QOS\_SLOWBW\_CONTROL\_13)**

Reset: 0000\_0000\_0000\_1000h.

QOS Slow Memory BW Control13

\_lthree[1:0]; MSRC000\_028D

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_028E (L3::L3CRB::L3QOS\_SLOWBW\_CONTROL\_14)**

Reset: 0000\_0000\_0000\_1000h.

QOS Slow Memory BW Control14

\_lthree[1:0]; MSRC000\_028E

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value**MSRC000\_028F (L3::L3CRB::L3QOS\_SLOWBW\_CONTROL\_15)**

Reset: 0000\_0000\_0000\_1000h.

QOS Slow Memory BW Control15

\_lthree[1:0]; MSRC000\_028F

**Bits Description**

63:13 Reserved.

12:0 **Ceiling.** Read-write. Reset: 1000h. QOS BW Control BW ceiling value

**MSRC000\_03FD (L3::L3CRB::L3\_QOS\_ABMC\_CFG)**

Reset: 0000\_0000\_0000\_007Fh.

L3 QOS ABMC Counter Config Register.

\_lthree[1:0]; MSRC000\_03FD

Bits	Description
63	<b>ConfigureCounter.</b> Read-write. Reset: 0. Configure the specified counter. Packed from MSR[63]
62	<b>EnableCounter.</b> Read-write. Reset: 0. Enable the specified counter. Packed from MSR[62]
61:53	Reserved.
52:48	<b>CounterId.</b> Read-write. Reset: 00h. ID of the counter to configure (or describe). Packed from MSR[52:48]
47	<b>BwSrcIsClos.</b> Read-write. Reset: 0. 0=RMID. 1=CLOS. Identifies if the BwSrc identifies an RMID or a CLOS. Packed from MSR[47]
46:44	Reserved.
43:32	<b>BwSrc.</b> Read-write. Reset: 000h. Which RMID (or CLOS) to track with the counter. Packed from MSR[43:32]
31:7	Reserved.
6	<b>L3CacheVicBwMon.</b> Read-write. Reset: 1. Dirty Victims from the QOS domain to all types of memory
5	<b>L3CacheRmtSlowBwFillMon.</b> Read-write. Reset: 1. Reads to slow memory in the non-local NUMA domain
4	<b>L3CacheLclSlowBwFillMon.</b> Read-write. Reset: 1. Reads to slow memory in the local NUMA domain
3	<b>L3CacheRmtBwNtWrMon.</b> Read-write. Reset: 1. Non-temporal writes to non-local NUMA domain
2	<b>L3CacheLclBwNtWrMon.</b> Read-write. Reset: 1. Non-temporal writes to local NUMA domain
1	<b>L3CacheRmtBwFillMon.</b> Read-write. Reset: 1. Reads to memory in the non-local NUMA domain
0	<b>L3CacheLclBwFillMon.</b> Read-write. Reset: 1. Reads to memory in the local NUMA domain

**MSRC000\_03FE (L3::L3CRB::L3\_QOS\_ABMC\_DSC)**

Read-only, Volatile. Reset: 0000\_0000\_0000\_007Fh.

L3 QOS ABMC Counter Config Register.

\_lthree[1:0]; MSRC000\_03FE

Bits	Description
63	<b>ConfigurationError.</b> Read-only, Volatile. Reset: 0. Configure the specified counter. Packed from MSR[63]
62	<b>EnableCounter.</b> Read-only, Volatile. Reset: 0. Enable the specified counter. Packed from MSR[62]
61:53	Reserved.
52:48	<b>CounterId.</b> Read-only, Volatile. Reset: 00h. ID of the counter to configure (or describe). Packed from MSR[52:48]
47	<b>BwSrcIsClos.</b> Read-only, Volatile. Reset: 0. 0=RMID. 1=CLOS. Identifies if the BwSrc identifies an RMID or a CLOS. Packed from MSR[47]
46:44	Reserved.
43:32	<b>BwSrc.</b> Read-only, Volatile. Reset: 000h. Which RMID (or CLOS) to track with the counter. Packed from MSR[43:32]
31:7	Reserved.
6	<b>L3CacheVicBwMon.</b> Read-only, Volatile. Reset: 1. Dirty Victims from the QOS domain to all types of memory
5	<b>L3CacheRmtSlowBwFillMon.</b> Read-only, Volatile. Reset: 1. Reads to slow memory in the non-local NUMA domain
4	<b>L3CacheLclSlowBwFillMon.</b> Read-only, Volatile. Reset: 1. Reads to slow memory in the local NUMA domain
3	<b>L3CacheRmtBwNtWrMon.</b> Read-only, Volatile. Reset: 1. Non-temporal writes to non-local NUMA domain
2	<b>L3CacheLclBwNtWrMon.</b> Read-only, Volatile. Reset: 1. Non-temporal writes to local NUMA domain
1	<b>L3CacheRmtBwFillMon.</b> Read-only, Volatile. Reset: 1. Reads to memory in the non-local NUMA domain
0	<b>L3CacheLclBwFillMon.</b> Read-only, Volatile. Reset: 1. Reads to memory in the local NUMA domain

**MSRC000\_03FF [L3 Qos Extended Configuration] (L3::L3CRB::L3QosExtCfg)**

Reset: 0000\_0000\_0000\_0000h.

AMD specific register for BW control (not x86 ISA specified)

\_lthree[1:0]; MSRC000\_03FF

Bits	Description
63:2	Reserved.
1	<b>CdmaToMaxCbm_En.</b> Read-write. Reset: 0. Enable ABMC
0	<b>ABMC_En.</b> Read-write. Reset: 0. Enable ABMC

**MSRC000\_0400 (L3::L3CRB::QOS\_EVT\_CFG\_0)**

Reset: 0000\_0000\_0000\_007Fh.

Identifies the Bandwidth Sources to include in Bandwidth Monitoring Event 0

\_lthree[1:0]; MSRC000\_0400

Bits	Description
63:7	Reserved.
6	<b>L3CacheVicBwMon.</b> Read-write. Reset: 1. Dirty Victims from the QOS domain to all types of memory
5	<b>L3CacheRmtSlowBwFillMon.</b> Read-write. Reset: 1. Reads to slow memory in the non-local NUMA domain
4	<b>L3CacheLclSlowBwFillMon.</b> Read-write. Reset: 1. Reads to slow memory in the local NUMA domain
3	<b>L3CacheRmtBwNtWrMon.</b> Read-write. Reset: 1. Non-temporal writes to non-local NUMA domain
2	<b>L3CacheLclBwNtWrMon.</b> Read-write. Reset: 1. Non-temporal writes to local NUMA domain
1	<b>L3CacheRmtBwFillMon.</b> Read-write. Reset: 1. Reads to memory in the non-local NUMA domain
0	<b>L3CacheLclBwFillMon.</b> Read-write. Reset: 1. Reads to memory in the local NUMA domain

**MSRC000\_0401 (L3::L3CRB::QOS\_EVT\_CFG\_1)**

Reset: 0000\_0000\_0000\_0015h.

Identifies the Bandwidth Sources to include in Bandwidth Monitoring Event 1

\_lthree[1:0]; MSRC000\_0401

Bits	Description
63:7	Reserved.
6	<b>L3CacheVicBwMon.</b> Read-write. Reset: 0. Dirty Victims from the QOS domain to all types of memory
5	<b>L3CacheRmtSlowBwFillMon.</b> Read-write. Reset: 0. Reads to slow memory in the non-local NUMA domain
4	<b>L3CacheLclSlowBwFillMon.</b> Read-write. Reset: 1. Reads to slow memory in the local NUMA domain
3	<b>L3CacheRmtBwNtWrMon.</b> Read-write. Reset: 0. Non-temporal writes to non-local NUMA domain
2	<b>L3CacheLclBwNtWrMon.</b> Read-write. Reset: 1. Non-temporal writes to local NUMA domain
1	<b>L3CacheRmtBwFillMon.</b> Read-write. Reset: 0. Reads to memory in the non-local NUMA domain
0	<b>L3CacheLclBwFillMon.</b> Read-write. Reset: 1. Reads to memory in the local NUMA domain

**MSRC001\_023[1...B] [L3 Performance Monitor Counter] (L3::L3CRB::ChL3Pmc)**

Read-write, Volatile. Reset: 0000\_0000\_0000\_0000h.

L3 Performance Monitor Counter Register

\_lthree[1:0]\_n0; MSRC001\_0231

\_lthree[1:0]\_n1; MSRC001\_0233

\_lthree[1:0]\_n2; MSRC001\_0235

\_lthree[1:0]\_n3; MSRC001\_0237

\_lthree[1:0]\_n4; MSRC001\_0239

\_lthree[1:0]\_n5; MSRC001\_023B

\_lthree0\_n[5:0]\_aliasSMN; L3L3CRBx20B7\_80[30,28,20,18,10,08]; L3L3CRB=0000\_0000h

\_lthree1\_n[5:0]\_aliasSMN; L3L3CRBx20F7\_80[30,28,20,18,10,08]; L3L3CRB=0000\_0000h

**Bits Description**

63:49 Reserved.

48 **Overflow.** Read-write, Volatile. Reset: 0. Counter overflow bit47:32 **CountHi.** Read-write, Volatile. Reset: 0000h. Bits 47:32 of the count31:0 **CountLo.** Read-write, Volatile. Reset: 0000\_0000h. Bits 31:0 of the count**MSRC001\_1095 [L3 Cache Range Reserve Base Address] (L3::L3CRB::L3RangeReserveBaseAddr)**

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]; MSRC001\_1095

**Bits Description**

63:52 Reserved.

51:12 **Addr.** Read-write. Reset: 00\_0000\_0000h. Base physical address bits 51:12 for the locked range.

11:0 Reserved.

**MSRC001\_1096 [L3 Cache Range Reserve Maximum Address] (L3::L3CRB::L3RangeReserveMaxAddr)**

Reset: 0000\_0000\_0000\_0000h.

\_lthree[1:0]; MSRC001\_1096

**Bits Description**

63:52 Reserved.

51:12 **Addr.** Read-write. Reset: 00\_0000\_0000h. Max physical address bits 51:12 for the locked range.

11:1 Reserved.

0 **En.** Read-write. Reset: 0. 0=Disable L3 Range Reservation. 1=Enable L3 Range Reservation. Enables the L3 range reservation when set.**MSRC001\_109A [L3 Cache Range Reservation Way Mask] (L3::L3CRB::L3RangeReserveWayMask)**

Reset: 0000\_0000\_0000\_0000h.

Way mask used to specify which L3 cache ways are used for range reservation

\_lthree[1:0]; MSRC001\_109A

**Bits Description**

63:16 Reserved.

15:0 **Mask.** Read-write. Reset: 0000h. L3 ways used for range reservation.**2.2.2 L3 Clocks and Test (CT) MSR Registers.**

**MSRC001\_0299 (L3::L3CT::L3RAPLPowerUnit0)**

Read-only. Reset: 0000\_0000\_0000\_0000h.

L3 RAPL Power Unit 0

\_lthree[1:0]; MSRC001\_0299

Bits	Description
63:20	Reserved.
19:16	<b>TimeUnits.</b> Read-only. Reset: 0h. Time information (in Seconds) is based on the multiplier, $1/2^{TU}$ where TU is unsigned. Default value is 1010b, indicating time unit is in 976 microseconds increment
15:13	Reserved.
12:8	<b>EnergyStatusUnits.</b> Read-only. Reset: 00h. Energy information (in Joules) is based on the multiplier, $1/2^{ESU}$ where ESU is unsigned integer. Default value is 10000b, indicating energy status unit is in 15.3 micro-Joules increment
7:4	Reserved.
3:0	<b>PowerUnits.</b> Read-only. Reset: 0h. Power information (in Watts) is based on the multiplier, $1/2^{PU}$ where PU is an unsigned integer. Default value is 0011b, indicating power unit is in 1/8 Watts increment

**MSRC001\_029B (L3::L3CT::L3CCXEnergyStatus)**

Read-only. Reset: 0000\_0000\_0000\_0000h.

L3 CCX Energy Status 0

\_lthree[1:0]; MSRC001\_029B

Bits	Description
63:0	<p><b>TotalEnergyConsumed.</b> Read-only. Reset: 0000_0000_0000_0000h.</p> <p><b>Description:</b> Total Energy consumed since the last time the register is cleared. It reports the actual energy use for respective power domain. This MSR is updated every ~1ms. Energy status is free running. Users calculate power for a given domain by calculating <math>dEnergy/dTime</math> for that domain. Users must ensure successive reads contain atleast one, but preferably many energy status updates by hardware. Readable/writable field for use by SMU.</p>

### 3 Reliability, Availability, and Serviceability (RAS) Features

A full implementation of RAS involves capabilities and support from the processor design, board hardware design, BIOS, firmware, and software.

#### 3.1 Machine Check Architecture

*Table 21: Machine Check Terms and Acronyms*

Term	Description
<b>MCA</b>	Machine Check Architecture.
<b>MCAX</b>	Machine Check Architecture eXtensions.
<b>WRIG</b>	Writes Ignored.

##### 3.1.1 Overview

The processor contains logic and registers to detect, log, and correct errors in the data or control paths. The Machine Check Architecture (MCA) defines facilities by which processor and system hardware errors are logged and reported to system software. This allows system software to perform a strategic role in recovery from and diagnosis of hardware errors.

##### 3.1.1.1 Legacy Machine Check Architecture

The legacy x86 Machine Check Architecture (MCA) refers to the standard x86 facilities for error logging and reporting. Refer to the AMD64 Architecture Programmer's Manual for an architectural overview of the Machine Check Architecture.

Support for the MCA is indicated by Core::X86::Cpuid::FeatureIdEdx[MCA] or Core::X86::Cpuid::FeatureExtIdEdx[MCA].

##### 3.1.1.2 Machine Check Architecture Extensions

Machine Check Architecture Extensions (MCAX) is AMD's x86-64 extension to the Machine Check Architecture.

Goals of MCAX include:

- Accommodate a variety of implementations, where each implementation may have a different assignment of MCA bank to block.
  - For example, one implementation may have 1 memory channel with an MCA bank, and another otherwise identical implementation may have 2 memory channels, each with their own MCA bank. Therefore, MCA bank allocation will appear different between these two implementations. MCAX is designed to require no assumptions about which MCA banks access which blocks.
  - Provide granular information for error logging, to improve error handling and diagnosability.
  - Preserve compatibility with system software which is not MCAX-aware.

Features of the MCA Extensions include:

- Increased MCA Bank Count: Features to support an expansion of the number of MCA banks supported by AMD processors.
- MCA Extension Registers: Expanded information logged in MCA banks to allow for improved error handling, better diagnosability, and future scalability.
- MCA DOER/SEER Roles: Separation of MCA information to take advantage of emerging software roles, namely

Error Management (Dynamic Operational Error Handling, or DOER) for managing running programs, and Fault Management (Symptom Elaboration of Errors, or SEER) for hardware diagnosability and reconfiguration. This clearer separation is accompanied by the assurances of architectural state (vs. implementation dependent state), so that operating systems can rely on the state and exploit new functionality.

Support for Machine Check Architecture Extensions (MCAX) is indicated by `Core::X86::Cpuid::RasCap[ScalableMca]`.

### 3.1.1.3 Use of MCA Information

The MCA registers contain information that can be used for multiple purposes. Some of this information is architecturally specified, and remains consistent from generation to generation, enabling portable, stable code. Some of this information is implementation specific; it is vital for diagnosis and other software functions, but may change with new implementations. It is important to understand how this information is categorized, and how it should be used. This section describes a framework for that.

There are two fundamental roles to be carried out after an error occurs; Error Management and Fault Management. All information required for Error Management is architectural and stable; some information required for Fault Management is also architectural.

#### 3.1.1.3.1 Error Management

Error Management describes actions necessary by operational software (e.g., the operating system or the hypervisor) to manage running programs that are affected by the error. The list of possible actions for operational error management is generally fairly short: take no action; terminate a single affected process, program, or virtual machine; terminate system operation. The Error Management role is defined as the DOER role (Dynamic Operational Error Handling). The name is intended to indicate an active role in managing running programs. Information used by the DOER is fairly limited and straightforward. It includes only those status fields needed to make decisions about the scope and severity of the error, and to determine what immediate action is to be taken.

#### 3.1.1.3.2 Fault Management

Fault Management describes optional actions for purposes of diagnosis, repair, and reconfiguration of the underlying hardware. The Fault Management role is described as SEER (Symptom Elaboration of Errors) because it peers further into hardware behavior and may try to influence future behavior via Predictive Fault Analysis, reconfiguration, service actions, etc. Because the SEER depends on understanding specifics of hardware configuration, it necessarily requires implementation specific knowledge and may not be portable across implementations.

Fields that are not explicitly specified as DOER are SEER. By separating error handling software into DOER and SEER roles, programmers can create both simpler and more functional code. The terms DOER and SEER appear in other sections of this document as an aid to reasoning about error handling and understanding actions to be taken.

### 3.1.2 Machine Check Registers

Host software references MCA registers via MSRs. MSRs are accessed through x86 WRMSR and RDMSR instructions. MSR addresses are private to a logical core; a given MSR referenced by two different cores results in references to two different MCA registers.

#### 3.1.2.1 Global Registers

`Core::X86::Cpuid::FeatureIdEdx[MCA]` or `Core::X86::Cpuid::FeatureExtIdEdx[MCA]` indicates the presence of the following machine check registers:

- Core::X86::Msr::MCG\_CAP
  - Reports how many machine check register banks are supported. This value reflects the number of MCA banks visible to that logical core. Some banks may be RAZ/WRIG either due to the bank being reserved or unused on this processor or because the block's MCA bank is controlled by another logical core.
- Core::X86::Msr::MCG\_STAT
  - Provides basic information about processor state after the occurrence of a machine check error.
- Core::X86::Msr::MCG\_CTL
  - Used by software to enable or disable the logging and reporting of machine check errors in the error reporting banks. Some bits may be RAZ/WRIG either due to the bank being reserved or unused on this processor or because the block's MCA bank is controlled by another logical core.
- Core::X86::Msr::McaIntrCfg
  - Used by software to configure certain machine check interrupts.

### 3.1.2.2 Machine Check Banks

A processor contains multiple blocks, and some of them have banks of machine check architecture registers (MCA banks). An MCA bank logs and reports errors to software.

The legacy MCA supports up to 32 MCA banks per logical core. MCAX supports up to 64 MCA banks per logical core.

The processor ensures that non-zero error status in an MCA bank is visible to exactly one logical core in a system, and that error notifications are directed to that logical core. Hardware also makes MCA bank configuration and control registers available to exactly one logical core. Banks associated with a CPU core are controlled by that logical core. Banks associated with other blocks are controlled by an implementation-specific logical core.

#### 3.1.2.2.1 Legacy MCA Registers

Each legacy MCA bank allocates address space for 4 legacy MCA registers.

The legacy MCA registers include:

- MCA\_CTL
  - Enables error reporting via machine check exception.
- MCA\_STATUS
  - Logs information associated with errors.
- MCA\_ADDR
  - Logs address information associated with errors.
- MCA\_MISC0
  - Logs miscellaneous information associated with errors.

#### 3.1.2.2.2 Legacy MCA MSRs

The legacy MCA MSRs are MSR0000\_04[7F:00]. The legacy MCA MSR space contains 32 banks of 4 registers per bank. The layout of the legacy MCA MSR space is given in Table 22 [Legacy MCA MSR Layout].

Table 22: Legacy MCA MSR Layout

MCA bank (decimal)	MCA_CTL (MSR0000_0xxx)	MCA_STATUS	MCA_ADDR	MCA_MISC0
0	400	401	402	403
1	404	405	406	407
2	408	409	40A	40B
3	40C	40D	40E	40F



4	410	411	412	413
5	414	415	416	417
6	418	419	41A	41B
...				
31	47C	47D	47E	47F

Features and registers associated with the MCA Extensions are not available in this legacy MSR address range. AMD recommends that operating systems use the MCAX MSR address range, rather than rely on the legacy MCA MSR address range.

All unimplemented or unused registers in the legacy MCA MSR address range are RAZ/WRIG. MC4 registers (MSR0000\_0410:0000\_0413) are RAZ/WRIG.

MSR0000\_0000 is aliased to the MCAX MSR address for MC0\_ADDR, and MSR0000\_0001 is aliased to the MCAX MSR address of MC0\_STATUS.

### 3.1.2.2.3 MCAX Registers

Each MCAX bank allocates address space for 16 MCA registers. All unimplemented registers in the MCA MSR space are RAZ/WRIG. MCAX bank registers include the legacy MCA registers as well as registers associated with the MCA Extensions.

The MCA Extension registers include:

- MCA\_CONFIG
  - Provide configuration capabilities for this MCA bank.
- MCA\_IPID
  - Provides information on the block associated with this MCA bank.
- MCA\_SYND
  - Logs physical location information associated with a logged error.
- MCA\_DESTATUS
  - Logs status information associated with a deferred error.
- MCA\_DEADDR
  - Logs address information associated with a deferred error.
- MCA\_MISC[1:4]
  - Provides additional threshold counters within an MCA bank.
- MCA\_SYND1 & MCA\_SYND2
  - Log information associated with a logged error, such as FruText.

### 3.1.2.2.4 MCAX MSRs

MCAX MSRs are present at MSRC000\_2[3FF:000]. This MSR address range contains space for 64 banks of 16 registers each. MSRC000\_2[FFF:400] are Reserved for future use. The MCAX MSR address range allows access to both legacy MCA registers and MCAX registers in each MCA bank.

The x86 MCAX MSR address format is SSSS\_SBBR (hex). S=MCA register space (i.e., MSRC000\_2XXX). B=MCA bank. R=Register offset within MCA bank. The layout of the MCAX MSR space is given in Table 23 [MCAX MSR Layout].

Access to unused MCAX MSRs is RAZ/WRIG. MCA Bank 4 is always Read-as-zero (RAZ/WRIG).

Table 23: MCAX MSR Layout

MCA bank	MCAX MSR (MSRC000_2xxx)												
	Legacy MCA Bank registers				MCAX Bank registers								
	CTL	STATUS	ADDR	MISC0	CONFIG	IPID	SYND	Reserved	DESTAT	DEADDR	MISC[4:1]	SYND1	SYND2
0	000	001	002	003	004	005	006	007	008	009	00D:00A	00E	00F
1	010	011	012	013	014	015	016	017	018	019	01D:01A	01E	01F
2	020	021	022	023	024	025	026	027	028	029	02D:02A	02E	02F
...													
63	3F0	3F1	3F2	3F3	3F4	3F5	3F6	3F7	3F8	3F9	3FD:3FA	3FE	3FF

All processors maintain the same mapping of MSR to MCA bank number (MSRC000\_2000 for the beginning of MCA Bank 0, MSRC000\_2010 for the beginning of MCA Bank 1, etc.), regardless of what block the bank represents (see 3.1.5.5 [Determining Bank Type]).

MCA\_CTL\_MASK MSRs are present at MSRC001\_04[3F:00]. MSRC001\_04[FF:40] are Reserved for future use. The layout of these registers is given in Table 24 [MCAX Implementation-Specific Register Layout].

Table 24: MCAX Implementation-Specific Register Layout

MCA bank	MCA_CTL_MASK (MSRC001_04xx)
0	00
1	01
2	02
...	
63	3F

### 3.1.2.3 Access Permissions

When McStatusWrEn == 0, a Write to an implemented MCA\_STATUS register causes a General Protection Fault (#GP) unless the value being written is zero. When McStatusWrEn == 1, a Write to an implemented MCA\_STATUS register does not cause a #GP regardless of data value.

Access to legacy MCA\_CTL\_MASK (MSRC001\_00xx) causes a General Protection Fault (#GP).

Access to legacy MC4\_MISC1-8 (MSRC000\_0408:C000\_040F) is RAZ/WRIG.

### 3.1.3 Machine Check Errors

#### 3.1.3.1 Error Severities

The classes of machine check errors are, in priority order from highest to lowest:

- Uncorrected
- Deferred
- Corrected

Uncorrected errors cannot be corrected by hardware. Uncorrected errors update the status and address registers if not masked from logging in MCA\_CTL\_MASK. Information in the status and address registers from a previously logged lower priority error is overwritten. Previously logged errors of the same priority are not overwritten. Uncorrected errors that are enabled for reporting in MCA\_CTL result in reporting to software via machine check exceptions. If an uncorrected error is masked from logging, the error is ignored by hardware (exceptions are noted in the register definitions). If an uncorrected error is disabled from reporting, containment of the error and logging/reporting of subsequent errors may be affected. Therefore, enable reporting of unmasked uncorrected errors for normal operation. Disable reporting of uncorrected errors only for debug purposes.

Deferred errors are errors that cannot be corrected by hardware, but do not cause an immediate interruption in program flow, loss of data integrity, or corruption of processor state. These errors indicate that data has been corrupted but not consumed; no exception is generated because the data has not been referenced by a core or an IO link. Hardware writes information to the status and address registers in the corresponding bank that identifies the source of the error if deferred errors are enabled for logging. If there is information in the status and address registers from a previously logged lower priority error, it is overwritten. Previously logged errors of the same or higher priority are not overwritten. Deferred errors are not reported via machine check exceptions; they can optionally be reported via LVT or SMI.

Corrected errors are those which have been corrected by hardware and cause no loss of data or corruption of processor state. Hardware writes the status and address registers in the corresponding register bank with information that identifies the source of the error if they are enabled for logging. Corrected errors are not reported via machine check exceptions. Some corrected errors may optionally be reported to software via LVT or SMI if the number of errors exceeds a configurable threshold.

An error to be logged when the status register contains valid data can result in an overflow condition. During error overflow conditions, the new error may not be logged or an error which has already been logged in the status register may be overwritten.

Table 25 [Error Overwrite Priorities] indicates which errors are overwritten in the error status registers.

*Table 25: Error Overwrite Priorities*

		Older Error		
		Uncorrected	Deferred	Corrected
Newer	Uncorrected	-	Overwrite	Overwrite
	Deferred	-	-	Overwrite

Error	Corrected	-	-	-
-------	-----------	---	---	---

Table 26 [Error Scope Hierarchy] provides a hierarchy of error scopes that determine the potential ability to recover the system based on fields in MCA\_STATUS when MCA\_STATUS[Val] == 1.

*Table 26: Error Scope Hierarchy*

PCC	UC	TCC	Deferred	Comments
1	X	X	X	Uncorrected system fatal error. Action required. A hardware-uncorrected error has corrupted system state. The error is fatal to the system and the system processing must be terminated.
0	1	1	X	Uncorrected thread fatal error. Action required. A hardware-uncorrected error has corrupted state for the process thread executing on the interrupted logical core. State for other process threads is unaffected.
0	1	0	X	Uncorrected recoverable error. Action required. A hardware-uncorrected error has not corrupted state of the process thread. Recovery of the process thread is possible if the uncorrected error is corrected by software.
0	0	0	1	Deferred error. Action optional. A hardware-uncorrected error has been discovered but not yet consumed. Error handling software may attempt to correct this error, or prevent access by processes which map the data, or make the physical resource containing the data inaccessible.
0	0	0	0	Corrected error. Action optional. A hardware-corrected error has been corrected. No action is required by error handling software.

### 3.1.3.2 Exceptions and Interrupts

Some or all errors logged in the MCA may require an interrupt or exception to be signaled.

The processor supports the following x86 interrupt/exception types to be communicated to the x86 core in response to an error:

- Machine Check Exception (MCE)
- System Management Interrupt (SMI)
- APIC based interrupt (LVT)

MCEs can be architecturally precise, context-synchronous, or asynchronous. An MCE that sets Core::X86::Msr::MCG\_STAT[RIPV] = 1 and Core::X86::Msr::MCG\_STAT[EIPV] = 1 is precise and the program can be restarted reliably. Other interrupts are architecturally asynchronous.

The ability of hardware to generate a machine check exception upon an error is indicated by Core::X86::Cpuid::FeatureIdEdx[MCE] or Core::X86::Cpuid::FeatureExtIdEdx[MCE].

### 3.1.3.3 Error Codes

The MCA\_STATUS[ErrorCode] field contains information used to identify the logged error. This section identifies how to decode the ErrorCode field.

*Table 27: Error Code Types*

Error Code	Error Code Type	Description
------------	-----------------	-------------

0000 0000 0001 TTLL	TLB	TT = Transaction Type LL = Cache Level
0000 0001 RRRR TTLL	Memory	RRRR = Memory Transaction Type TT = Transaction Type LL = Cache Level
0000 1XXT RRRR XXLL	Bus	XX = Reserved T = Timeout RRRR = Memory Transaction Type LL = Cache Level
0000 01UU 0000 0000	Internal Unclassified	UU = Internal Error Type

Table 28: Error code: transaction type (TT)

TT	Transaction Type
00	Instruction
01	Data
10	Generic
11	Reserved

Table 29: Error codes: cache level (LL)

LL	Cache Level
00	L0: Core
01	L1: Level 1
10	L2: Level 2
11	LG: Generic

Table 30: Error codes: memory transaction type (RRRR)

RRRR	Memory Transaction Type
0000	Generic
0001	Generic Read
0010	Generic Write
0011	Data Read
0100	Data Write
0101	Instruction Fetch
0110	Prefetch
0111	Evict
1000	Snoop (Probe)

Errors can also be identified by the MCA\_STATUS[ErrorCodeExt] field. MCA\_STATUS[ErrorCodeExt] indicates which bit position in the corresponding MCA\_CTL register enables error reporting for the logged error. For instance, MCA\_STATUS[ErrorCodeExt] == 0x9 means that the logged error is enabled by MCA\_CTL[9], and the description of MCA\_CTL[9] contains information on decoding the error log. Specific ErrorCodeExt values are implementation dependent, and should not be used by architectural or portable code.

### 3.1.3.4 Extended Error Codes

The MCA\_STATUS[ErrorCodeExt] field contains additional information used to identify the logged error. Error positions in MCA\_CTL and MCA\_CTL\_MASK and Extended Error Codes are fixed within a given bank type. That is, for an MCA bank with a given MCA\_IPID[HwId, McaType] value, the processor ensures that the same error is reported in a given bit

position of of MCA\_CTL regardless of the product in which that bank appears. Similarly, for an MCA bank with a given MCA\_IPID[HwId, McaType] value, hardware ensures that the mapping of errors to Extended Error Codes is consistent across products.

### 3.1.3.5 DOER and SEER State

The DOER fields are:

- MCG\_STAT
  - Count
  - MCIP
  - RIPV
  - EIPV
- MCA\_STATUS
  - Val
  - PCC
  - TCC
  - UC
  - MiscV
  - AddrV

The MCA\_STATUS[Deferred] bit is used for SEER functionality but is architectural.

### 3.1.3.6 MCA Overflow Recovery

MCA Overflow Recovery is a feature allowing recovery of the system when the overflow bit is set. MCA Overflow Recovery is supported when `Core::X86::Cpuid::RasCap[McaOverflowRecov] == 1`.

When MCA Overflow Recovery is supported, software may rely on `MCA_STATUS[PCC] == 1` to indicate all system-fatal conditions. When MCA Overflow Recovery is not supported, an uncorrected error logged with `MCA_STATUS[Overflow] = 1` may indicate the system-fatal condition that an error requiring software intervention was not logged. Therefore, software must terminate system processing whenever an uncorrected error is logged with `MCA_STATUS[Overflow] = 1`.

### 3.1.3.7 MCA Recovery

MCA Recovery is a feature allowing recovery of the system when the hardware cannot correct an error. MCA Recovery is supported when `Core::X86::Cpuid::RasCap[SUCCOR] == 1`.

When MCA Recovery is supported and an uncorrected error has been detected that the hardware can contain to the task or process to which the machine check has been delivered, it logs a context-synchronous uncorrectable error (`MCA_STATUS[UC] = 1`, `MCA_STATUS[PCC] = 0`). The rest of the system is unaffected and may continue running if supervisory software can terminate only the affected process or VM.

## 3.1.4 Machine Check Features

### 3.1.4.1 Error Thresholding

For some types of errors, the hardware maintains counts of the number of errors. When the counter reaches a programmable threshold, an event may optionally be triggered to signal system software. This is known as error

thresholding. The primary purpose of error thresholding is to help software recognize an excessive rate of errors, which may indicate marginal or failing hardware. This information can be used to make decisions about deconfiguring hardware or scheduling service actions. The error count is incremented for corrected, deferred, and uncorrected errors.

The MCA\_MISCx registers contain the architectural interface for error thresholding. The registers contain a 12-bit error counter that can be initialized to any value except FFFh, with the option to interrupt when the counter reaches FFFh.

MCA\_MISCx[ThresholdIntType] determines the type of interrupt to be generated for threshold overflow errors in that counter. This can be set to None, LVT, or SMI. If this is set to LVT, Core::X86::Msr::McaIntrCfg[ThresholdLvtOffset] specifies the LVT offset that is used. Only one LVT offset is used per socket and the interrupt is routed to the APIC of the logical core from which the MCA bank is visible.

### 3.1.4.2 Error Simulation

Error simulation involves creating the appearance to software that an error occurred, and can be used to debug machine check interrupt handlers. See Core::X86::Msr::HWCR[McStatusWrEn] for making MCA registers writable for non-zero values. When McStatusWrEn is set, privileged software can write non-zero values to the specified registers without generating exceptions, and then simulate a machine check using the INT18 instruction (INTn instruction with an operand of 18). Setting a reserved bit in these registers does not generate an exception when this mode is enabled. However, setting a reserved bit may result in undefined behavior.

### 3.1.5 Software Guidelines

#### 3.1.5.1 Recognizing MCAX Support

Software which reads the MCA registers must recognize whether an implementation uses the legacy format or the MCAX format. This is accomplished by starting with CPUID Fn8000\_0007\_EBX[ScalableMca]. If ScalableMca == 1, then the implementation supports the MCAX indicator (MCA\_CONFIG[Mcac]). An MCA bank is an MCAX bank if MCA\_CONFIG[Mcac] == 1 in that bank.

#### 3.1.5.2 Communicating MCAX Support

Software which supports MCAX must set MCA\_CONFIG[McacEn] = 1 in each MCA bank.

Software that supports MCAX should use the MCAX MSRs to access both legacy and MCAX registers.

#### 3.1.5.3 Machine Check Initialization

The following initialization sequence must be followed:

- Platform firmware must initialize the MCA\_CTL\_MASK registers prior to the initialization of the MCA\_CTL registers and Core::X86::Msr::MCG\_CTL. Platform firmware and the operating system must not clear MCA\_CTL\_MASK bits that are set to 1. MCA\_CTL\_MASK registers must be set the same across all cores.
- The operating system must initialize the MCA\_CONFIG registers prior to initialization of the MCA\_CTL registers.
- The MCA\_CTL registers must be initialized prior to enabling the error reporting banks in MCG\_CTL.
- The Core::X86::Msr::MCG\_CTL register must be programmed identically for all cores in a processor, although the Read-write bits may differ per core.
- CR4.MCE must be set to enable machine check exceptions.

The operating system should configure the MCA\_CONFIG registers as follows:

- MCA\_CONFIG[McaXEn] = 1 if the operating system has been updated to use the MCA Extension MSR addresses. Otherwise, the operating system should preserve the platform firmware-programmed value of this field.
- MCA\_CONFIG[LogDeferredInMcaStat] and MCA\_CONFIG[DeferredIntType] to appropriate values based on OS support for deferred errors.

MCA\_STATUS MSRs are cleared by hardware after a cold reset. If initializing after a warm reset, then platform firmware should check for valid MCA errors and if present save the status for later diagnostic use.

Platform firmware may initialize the MCA without setting CR4.MCE; this results in a shutdown on any machine check which would have caused a machine check exception (followed by a reboot if configured). Alternatively, platform firmware that wishes to ensure continued operation in the event that a machine check occurs during boot may write MCG\_CTL with all ones and write zeros into each MCA\_CTL register. With these settings, a machine check error results in MCA\_STATUS being written without generating a machine check exception or a shutdown. Platform firmware may then poll MCA\_STATUS registers during critical sections of boot to ensure system integrity. Note that the system may be operating with corrupt data before polling MCA\_STATUS registers. Before passing control to the operating system, platform firmware should restore the values of those registers to what the operating system is expecting.

After MCA initialization, system software should check the Val bit on each MCA\_STATUS register. It is possible that valid error status information has already been logged in the MCA\_STATUS registers at the time software is attempting to initialize them. The status can reflect errors logged prior to a warm reset or errors recorded during the system power-up and boot process. Before clearing the MCA\_STATUS registers, software should examine their contents and log any errors found.

#### 3.1.5.4 Determining Bank Count

System software should read Core::X86::Msr::MCG\_CAP[Count] to determine the number of machine check banks visible to a logical core. The banks are numbered from 0 to one less than the value found in Core::X86::Msr::MCG\_CAP[Count]. For example, if the Count field indicates five banks are supported, they are numbered MC0 through MC4.

#### 3.1.5.5 Determining Bank Type

To determine which type of block is mapped to an MCA bank, software can query the MCA\_IPID register within that bank. This register exists when MCA\_CONFIG[McaX] == 1 in a given bank.

MCA\_IPID[HardwareID] provides the block type for the block that contains this MCA bank. For blocks that contain multiple MCA bank types (e.g., CPU cores), MCA\_IPID[McaType] provides an identifier for the type of MCA bank. MCA\_IPID[McaType] values are specific to a given MCA\_IPID[HardwareID]. Therefore, an MCA bank type can be identified by the value of {MCA\_IPID[Hwid], MCA\_IPID[McaType]}. For instance, the CPU core's LS bank is identified by MCA::LS::MCA\_IPID\_LS[HardwareID] == 176 and MCA::LS::MCA\_IPID\_LS[McaType] == 0. An MCA\_IPID[HardwareID] value of 0 indicates an unpopulated MCA bank that is ensured to be RAZ/WRIG.

MCA\_IPID[InstanceId] provides a unique instance number to allow software to differentiate blocks with multiple identical instances within a processor. MCA\_IPID[InstanceId] values are processor-specific and are not ensured to be stable across different processor generations.

#### 3.1.5.6 Recognizing Error Type

Software can use the combination of MCA\_IPID[Hwid, McaType] and MCA\_STATUS[ErrorCodeExt] to recognize a specific error type.



### 3.1.5.7 Machine Check Error Handling

A machine check handler is invoked to handle an exception for a particular thread. The information needed by the machine check handler is not shared with other threads, so no cross-thread coordination or special handling is required. Specifically, all MCA banks are only visible from a single thread, so software on a single thread can access each bank through MSR space without contention from other threads.

At a minimum, the machine check handler must be capable of logging error information for later examination. The handler should log as much information as is needed to diagnose the error. More thorough exception handler implementations can analyze errors to determine if each error is recoverable by software. If a recoverable error is identified, the exception handler can attempt to correct the error and restart the interrupted program. An error may not be recoverable for the process or virtual machine it directly affects, but may be containable, so that other processes or virtual machines in the system are unaffected and system operation is recovered.

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

- Data collection:
  - Read Core::X86::Msr::MCG\_CAP[Count] to determine the number of status registers visible to the logical core.
  - All status registers in all error reporting banks must be examined to identify the cause of the machine check exception.
  - Check the valid bit in each status register (MCA\_STATUS[Val]). The remainder of the status register should be examined only when its valid bit is set.
  - When identifying the error condition and determining how to handle the error, portable exception handlers should examine only DOER fields in machine check registers.
  - Error handlers should collect all available MCA information, but should only interrogate details to the level which affects their actions. Lower level details may be useful for diagnosis and root cause analysis, but not for error handling.
  - Error handlers should save the values in MCA\_ADDR, MCA\_MISC0, and MCA\_SYND even if MCA\_STATUS[AddrV], MCA\_STATUS[MiscV], and MCA\_STATUS[SyndV] are zero. Error handlers should save the values in MCA\_MISC[4:1] if the registers exist.
- DOER Error Management:
  - Check MCA\_STATUS[PCC].
    - If PCC is set, error recovery is not possible. The handler should log the error information and terminate the system. If PCC is clear, the handler may continue with the following recovery steps.
  - Check MCA\_STATUS[UC].
    - If UC is set, the processor did not correct the error. Continue with the following recovery steps.
      - If MCA Overflow Recovery is not supported, and MCA\_STATUS[Overflow] == 1, error recovery is not possible; follow the steps for PCC = 1. See 3.1.3.6 [MCA Overflow Recovery].
      - If MCA Recovery is not supported, error recovery is not possible; follow the steps for PCC = 1. See 3.1.3.7 [MCA Recovery].
      - If MCA Recovery is supported:
        - Check MCA\_STATUS[TCC].
          - If TCC is set, the context of the process thread executing on the interrupted logical core may be corrupt and the thread cannot be recovered. The rest of the system is unaffected; it is possible to terminate only the affected process thread.
          - If TCC is clear, the context of the process thread executing on the interrupted logical core is not corrupt. Recovery of the process thread may be possible, but only if the uncorrected error condition is first corrected by software. Otherwise, the interrupted process thread must be terminated.

- Legacy exception handlers can check Core::X86::Msr::MCG\_STAT[RIPV] and Core::X86::Msr::MCG\_STAT[EIPV] in place of MCA\_STATUS[TCC]. If RIPV == EIPV == 1, the interrupted program can be restarted reliably. Otherwise, the program cannot be restarted reliably.
- If UC is clear, the processor either corrected or deferred the error and no software action is needed. The handler can log the error information and continue process execution.
- Exit:
  - When an exception handler is able to successfully log an error condition, clear the MCA\_STATUS registers prior to exiting the machine check handler.
  - Prior to exiting the machine check handler, clear Core::X86::Msr::MCG\_STAT[MCIP]. MCIP indicates that a machine check exception is in progress. If this bit is set when another machine check exception occurs, the processor enters the shutdown state.

## 3.2 Machine Check Architecture Implementation

### 3.2.1 Implemented Machine Check Banks

Table 31: Blocks Capable of Supporting MCA Banks

Acronym	Block Function
LS	Load-Store Unit
IF	Instruction Fetch Unit
L2	L2 Cache Unit
DE	Decode Unit
EX	Execution Unit
FP	Floating-Point Unit
L3	L3 Cache Unit
PIE	Power Management, Interrupts, Etc.
CS	Coherent Station
UMC	Unified Memory Controller
NBIO	Northbridge IO Unit

Table 32: Mapping of Blocks to MCA\_IPID[HwId] and MCA\_IPID[McaType]

Block	Hardware ID	MCA Type
LS	0xb0	0x0
IF	0xb0	0x1
L2	0xb0	0x2
L3	0xb0	0x7
UMC	0x96	0x0
PIE	0x2e	0x1
NBIO	0x18	0x0
CS	0x2e	0x2
EX	0xb0	0x5
FP	0xb0	0x6
DE	0xb0	0x3

### 3.2.2 Implemented Machine Check Bank Registers

Table 33 [Legacy MCA Registers] provides links to the description of each block's Legacy MCA registers. Table 34 [MCAX Registers] provides links to the description of each block's MCA Extension Registers.

*Table 33: Legacy MCA Registers*

Block	MCA Register				
	CTL	STATUS	ADDR	MISC	CTL_MASK
LS	MCA::LS::MCA_CTL_LS	MCA::LS::MCA_STATUS_LS	MCA::LS::MCA_ADDR_LS	MCA::LS::MCA_MISC0_LS	MCA::LS::MCA_CTL_MASK_LS
IF	MCA::IF::MCA_CTL_IF	MCA::IF::MCA_STATUS_IF	MCA::IF::MCA_ADDR_IF	MCA::IF::MCA_MISC0_IF	MCA::IF::MCA_CTL_MASK_IF
L2	MCA::L2::MCA_CTL_L2	MCA::L2::MCA_STATUS_L2	MCA::L2::MCA_ADDR_L2	MCA::L2::MCA_MISC0_L2	MCA::L2::MCA_CTL_MASK_L2
DE	MCA::DE::MCA_CTL_DE	MCA::DE::MCA_STATUS_DE	MCA::DE::MCA_ADDR_DE	MCA::DE::MCA_MISC0_DE	MCA::DE::MCA_CTL_MASK_DE
EX	MCA::EX::MCA_CTL_EX	MCA::EX::MCA_STATUS_EX	MCA::EX::MCA_ADDR_EX	MCA::EX::MCA_MISC0_EX	MCA::EX::MCA_CTL_MASK_EX
FP	MCA::FP::MCA_CTL_FP	MCA::FP::MCA_STATUS_FP	MCA::FP::MCA_ADDR_FP	MCA::FP::MCA_MISC0_FP	MCA::FP::MCA_CTL_MASK_FP
L3	MCA::L3::MCA_CTL_L3	MCA::L3::MCA_STATUS_L3	MCA::L3::MCA_ADDR_L3	MCA::L3::MCA_MISC0_L3	MCA::L3::MCA_CTL_MASK_L3
PIE	MCA::PIE::MCA_CTL_PIE	MCA::PIE::MCA_STATUS_PIE	MCA::PIE::MCA_ADDR_PIE	MCA::PIE::MCA_MISC0_PIE	MCA::PIE::MCA_CTL_MASK_PIE
CS	MCA::CS::MCA_CTL_CS	MCA::CS::MCA_STATUS_CS	MCA::CS::MCA_ADDR_CS	MCA::CS::MCA_MISC0_CS	MCA::CS::MCA_CTL_MASK_CS
UMC	MCA::UMC::MCA_CTL_UMC	MCA::UMC::MCA_STATUS_UMC	MCA::UMC::MCA_ADDR_UMC	MCA::UMC::MCA_MISC0_UMC MCA::UMC::MCA_MISC1_UMC	MCA::UMC::MCA_CTL_MASK_UMC
NBIO	MCA::NBIO::MCA_CTL_NBIO	MCA::NBIO::MCA_STATUS_NBIO	MCA::NBIO::MCA_ADDR_NBIO	MCA::NBIO::MCA_MISC0_NBIO	MCA::NBIO::MCA_CTL_MASK_NBIO

*Table 34: MCAX Registers*

Block	MCA Register				
	CONFIG	IPID	SYND	DESTAT	DEADDR
LS	MCA::LS::MCA_CONFIG_LS	MCA::LS::MCA_IPID_LS	MCA::LS::MCA_SYND_LS	MCA::LS::MCA_DESTAT_LS	MCA::LS::MCA_DEADDR_LS
IF	MCA::IF::MCA_CONFIG_IF	MCA::IF::MCA_IPID_IF	MCA::IF::MCA_SYND_IF	--	--
L2	MCA::L2::MCA_CONFIG_L2	MCA::L2::MCA_IPID_L2	MCA::L2::MCA_SYND_L2	MCA::L2::MCA_DESTAT_L2	MCA::L2::MCA_DEADDR_L2
DE	MCA::DE::MCA_CONFIG_DE	MCA::DE::MCA_IPID_DE	MCA::DE::MCA_SYND_DE	--	--
EX	MCA::EX::MCA_CONFIG_EX	MCA::EX::MCA_IPID_EX	MCA::EX::MCA_SYND_EX	--	--
FP	MCA::FP::MCA_CONFIG_FP	MCA::FP::MCA_IPID_FP	MCA::FP::MCA_SYND_FP	--	--
L3	MCA::L3::MCA_CONFIG_L3	MCA::L3::MCA_IPID_L3	MCA::L3::MCA_SYND_L3	MCA::L3::MCA_DESTAT_L3	MCA::L3::MCA_DEADDR_L3
PIE	MCA::PIE::MCA_CONFIG_PIE	MCA::PIE::MCA_IPID_PIE	MCA::PIE::MCA_SYND_PIE	MCA::PIE::MCA_DESTAT_PIE	MCA::PIE::MCA_DEADDR_PIE
CS	MCA::CS::MCA_CONFIG_CS	MCA::CS::MCA_IPID_CS	MCA::CS::MCA_SYND_CS	MCA::CS::MCA_DESTAT_CS	MCA::CS::MCA_DEADDR_CS
UMC	MCA::UMC::MCA_CONFIG_UMC	MCA::UMC::MCA_IPID_UMC	MCA::UMC::MCA_SYND_UMC	MCA::UMC::MCA_DESTAT_UMC	MCA::UMC::MCA_DEADDR_UMC
NBIO	MCA::NBIO::MCA_CONFIG_NBIO	MCA::NBIO::MCA_IPID_NBIO	MCA::NBIO::MCA_SYND_NBIO	MCA::NBIO::MCA_DESTAT_NBIO	MCA::NBIO::MCA_DEADDR_NBIO

### 3.2.3 Mapping of Banks to Blocks

Table 35 [Core MCA Bank to Block Mapping] shows MCA banks that are present in the address space of every logical core.

*Table 35: Core MCA Bank to Block Mapping*

Bank	Block
0	LS
1	IF
2	L2
3	DE
4	RAZ
5	EX
6	FP

Table 36 [Non-core MCA Bank to Block Mapping] shows MCA banks that are present in the address space of specific logical cores.

*Table 36: Non-core MCA Bank to Block Mapping*

Bank	Thread 0	Thread 2
0	LS	LS
1	IF	IF
2	L2	L2
3	DE	DE
4	RAZ	RAZ
5	EX	EX
6	FP	FP
7	L3	L3
8	L3	L3
9	L3	L3
10	L3	L3
11	L3	L3
12	L3	L3
13	L3	L3
14	L3	L3
15	UMC	RAZ
16	UMC	RAZ
17	UMC	RAZ
18	UMC	RAZ
19	CS	RAZ
20	CS	RAZ
21	CS	RAZ
22	CS	RAZ
23	RAZ	RAZ
24	RAZ	RAZ
25	RAZ	RAZ
26	RAZ	RAZ

27	PIE	RAZ
28	NBIO	RAZ
29	RAZ	RAZ
30	RAZ	RAZ
31	RAZ	RAZ
32	RAZ	RAZ
33	RAZ	RAZ
34	RAZ	RAZ
35	RAZ	RAZ
36	RAZ	RAZ
37	RAZ	RAZ
38	RAZ	RAZ
39	RAZ	RAZ
40	RAZ	RAZ
41	RAZ	RAZ
42	RAZ	RAZ
43	RAZ	RAZ
44	RAZ	RAZ
45	RAZ	RAZ
46	RAZ	RAZ
47	RAZ	RAZ
48	RAZ	RAZ
49	RAZ	RAZ
50	RAZ	RAZ
51	RAZ	RAZ
52	RAZ	RAZ
53	RAZ	RAZ
54	RAZ	RAZ
55	RAZ	RAZ
56	RAZ	RAZ
57	RAZ	RAZ
58	RAZ	RAZ
59	RAZ	RAZ
60	RAZ	RAZ
61	RAZ	RAZ
62	RAZ	RAZ
63	RAZ	RAZ

### 3.2.4 Decoding Error Type

If a valid error is logged in MCA\_STATUS or MCA\_DESTAT of an MCA bank:

1. Read the values of this bank's MCA\_IPID and MCA\_STATUS registers.
2. Use Table 31 [Blocks Capable of Supporting MCA Banks] to look up the block associated with the values of MCA\_IPID[HwId] and MCA\_IPID[McaType].
3. In 3.2.5 [MCA Banks], find the sub-section associated with the block in error.
4. In this sub-section, find the MCA\_STATUS table.
5. In the table, look up the row associated with the MCA\_STATUS[ErrorCodeExt] value.
6. The error type in this row is the logged error. The MCA\_STATUS, MCA\_ADDR and MCA\_SYND tables contain

information associated with this error.

7. If there is an error in both MCA\_STATUS and MCA\_DESTAT, the registers contain the same error if MCA\_STATUS[Deferred] is set. If MCA\_STATUS[Deferred] is not set, MCA\_DESTAT contains information for a different error than MCA\_STATUS.

### 3.2.5 MCA Banks

#### 3.2.5.1 LS

##### MSR0000\_0400...MSRC000\_2000 [LS Machine Check Control Thread 0] (MCA::LS::MCA\_CTL\_LS)

Read-write. Reset: 0000\_0000\_0000\_0000h.

0=Disables error reporting for the corresponding error. 1=Enables error reporting via machine check exception for the corresponding error. The MCA::LS::MCA\_CTL\_LS register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG\_CTL. Does not affect error detection, correction, or logging.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst0\_n[23:0]\_aliasMSRLEGACY; MSR0000\_0400

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst0\_n[23:0]\_aliasMSR; MSRC000\_2000

Bits	Description
63:27	Reserved.
26	<b>Hwa</b> . Read-write. Reset: 0. Hardware Asserts (HWAs)
25	<b>SystemReadDataErrorWcb</b> . Read-write. Reset: 0. System Read Data Error detected by write combine buffer. An error in a read of a line from the data fabric. Possible reasons include decode error and target abort.
24	<b>ScbDataNonCacheable</b> . Read-write. Reset: 0. Error on SCB data for non-cacheable DRAM or IO, discovered at data-pull time
23	<b>ScbData1</b> . Read-write. Reset: 0. Error on SCB data, commit pipe 1, discovered at SCB commit time
22	<b>ScbData0</b> . Read-write. Reset: 0. Error on SCB data, commit pipe 0, discovered at SCB commit time
21	<b>ScbStateAddr</b> . Read-write. Reset: 0. Error on SCB cacheline state (way and moesi state) or address field
20	<b>L2DataErr</b> . Read-write. Reset: 0. L2 Fill Data error.
19	<b>DcTagErr7</b> . Read-write. Reset: 0. DC Tag error type 5.
18	<b>DcTagErr3</b> . Read-write. Reset: 0. DC Tag error type 3.
17	<b>PDC</b> . Read-write. Reset: 0. PDC parity error.
16	<b>L2DTLB</b> . Read-write. Reset: 0. Level 2 TLB parity error.
15	<b>DcTagErr4</b> . Read-write. Reset: 0. DC Tag error type 4.
14	<b>DcDataErr3</b> . Read-write. Reset: 0. DC Data error type 3.
13	<b>DcDataErr2</b> . Read-write. Reset: 0. DC Data error type 2.
12	<b>DcDataErr1</b> . Read-write. Reset: 0. DC Data error type 1 and poison consumption. MCA_STATUS[Poison] is set on poison consumption from L2/L3.
11	<b>DcTagErr2</b> . Read-write. Reset: 0. DC Tag error type 2.
10	<b>SystemReadDataErrorMab</b> . Read-write. Reset: 0. System Read Data Error detected by mab. An error in a read of a line from the data fabric. Possible reasons include decode error and target abort.
9	<b>SystemReadDataErrorUcode</b> . Read-write. Reset: 0. System Read Data Error logged by ucode. An error in a read of a line from the data fabric. Possible reasons include decode error and target abort.
8	<b>IntErrTyp2</b> . Read-write. Reset: 0. Internal error type 2.
7	<b>IntErrTyp1</b> . Read-write. Reset: 0. Internal error type 1.
6	<b>DcTagErr1</b> . Read-write. Reset: 0. DC Tag error type 1.
5	<b>DcTagErr6</b> . Read-write. Reset: 0. DC Tag error type 6.
4	<b>DcTagErr5</b> . Read-write. Reset: 0. DC Tag error type 5.
3	<b>L1DTLB</b> . Read-write. Reset: 0. Level 1 TLB parity error.
2	<b>MAB</b> . Read-write. Reset: 0. Miss address buffer payload parity error.
1	<b>STQ</b> . Read-write. Reset: 0. Store queue parity error.
0	<b>LDQ</b> . Read-write. Reset: 0. Load queue parity error.

**MSR0000\_0001...MSRC000\_2001 [LS Machine Check Status Thread 0] (MCA::LS::MCA\_STATUS\_LS)**

Reset: Cold,0000\_0000\_0000\_0000h.

Logs information associated with errors.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst0\_n[23:0]\_aliasMSRSLLEGACY; MSR0000\_0001

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst0\_n[23:0]\_aliasMSRLEGACY; MSR0000\_0401

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst0\_n[23:0]\_aliasMSR; MSRC000\_2001

Bits	Description
63	<b>Val.</b> Reset: Cold,0. 1=A valid error has been detected. This bit should be cleared by software after the register has been read. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
62	<b>Overflow.</b> Reset: Cold,0. 1=An error was detected while the valid bit (Val) was set; at least one error was not logged. Overflow is set independently of whether the existing error is overwritten. See 3.1.3 [Machine Check Errors]. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
61	<b>UC.</b> Reset: Cold,0. 1=The error was not corrected by hardware. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
60	<b>En.</b> Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::LS::MCA_CTL_LS. This bit is a copy of bit in MCA::LS::MCA_CTL_LS for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	<b>MiscV.</b> Reset: Cold,0. 1=Valid thresholding in MCA::LS::MCA_MISC0_LS. In certain modes, MISC registers are owned by platform firmware and will RAZ when read by non-SMM code. Therefore, it is possible for MiscV=1 and the MISC register to read as all zeros. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
58	<b>AddrV.</b> Reset: Cold,0. 1=MCA::LS::MCA_ADDR_LS contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	<b>PCC.</b> Reset: Cold,0. 1=Hardware context held by the processor may have been corrupted. Continued operation of the system may have unpredictable results. The error is not recoverable or survivable, and the system should be reinitialized. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
56	<b>ErrCoreIdVal.</b> Reset: Cold,0. 1=The ErrCoreId field is valid. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
55	<b>TCC.</b> Reset: Cold,0. 1=Hardware context of the process thread to which the error was reported may have been corrupted. Continued operation of the thread may have unpredictable results. The thread must be terminated. Only meaningful when MCA::LS::MCA_STATUS_LS[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	<b>RESERV54.</b> Reset: Cold,0. MCA_STATUS Register Reserved bit. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
53	<b>SyndV.</b> Reset: Cold,0. 1=This error logged information in MCA::LS::MCA_SYND_LS. If MCA::LS::MCA_SYND_LS[ErrorPriority] is the same as the priority of the error in MCA::LS::MCA_STATUS_LS, then the information in MCA::LS::MCA_SYND_LS is associated with the error in MCA::LS::MCA_STATUS_LS. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	<b>RESERV47.</b> Reset: Cold,00h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
46	<b>CECC.</b> Reset: Cold,0. 1=The error was a correctable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
45	<b>UECC.</b> Reset: Cold,0. 1=The error was an uncorrectable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor.



	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
44	<b>Deferred.</b> Reset: Cold,0. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; an exception is deferred until the erroneous data is consumed.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
43	<b>Poison.</b> Reset: Cold,0. 1=The error was the result of attempting to consume poisoned data.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
42:41	<b>RESERV41.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
40	<b>Scrub.</b> Reset: Cold,0. 1=The error was the result of a scrub operation.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
39:38	<b>RESERV38.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
37:32	<b>ErrCoreId.</b> Reset: Cold,00h. When ErrCoreIdVal=1 this field indicates which core within the processor is associated with the error; Otherwise this field is reserved.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
31:30	<b>RESERV30.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
29:24	<b>AddrLsb.</b> Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::LS::MCA_ADDR_LS[ErrorAddr]. A value of 0 indicates that MCA::LS::MCA_ADDR_LS[63:0] contains a valid byte address. A value of 6 indicates that MCA::LS::MCA_ADDR_LS[63:6] contains a valid cache line address and that MCA::LS::MCA_ADDR_LS[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::LS::MCA_ADDR_LS[63:12] contain a valid 4KB memory page and that MCA::LS::MCA_ADDR_LS[11:0] should be ignored by error handling software.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
23:22	<b>RESERV22.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
21:16	<b>ErrorCodeExt.</b> Reset: Cold,00h. Extended Error Code. This field is used to identify the error type for root cause analysis. This field indicates which bit position in MCA::LS::MCA_CTL_LS enables error reporting for the logged error.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
15:0	<b>ErrorCode.</b> Reset: Cold,0000h. Error code for this error. See 3.1.3.3 [Error Codes] for details on decoding this field.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.

Table 37: MCA\_STATUS\_LS

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
LDQ	0x0	1	1	1	0	0	0
STQ	0x1	1	1	1	0	0	0
MAB	0x2	1	1	1	0	0	0
L1DTLB	0x3	1	1	1	0	0	1
DcTagErr5	0x4	1	1	1	0	0	0
DcTagErr6	0x5	1	1	1	0	0	0
DcTagErr1	0x6	1	1	1	0	0	0
IntErrTyp1	0x7	1	1	1	0	0	0
IntErrTyp2	0x8	0/1	0/1	0/1	0	0	0
SystemRead DataErrorUc	0x9	1	1	1	0	0	0/1

ode							
SystemRead DataErrorMa b	0xa	1	1	1	0	0	0/1
DcTagErr2	0xb	0	0	0	0	0	0
DcDataErr1	0xc	0/1	0	0/1	0	0/1	1
DcDataErr2	0xd	0	0	0	0/1	0	1
DcDataErr3	0xe	0	0	0	0/1	0	0/1
DcTagErr4	0xf	0	0	0	1	0	0
L2DTLB	0x10	0	0	0	0	0	0/1
PDC	0x11	0	0	0	0	0	0/1
DcTagErr3	0x12	0	0	0	0	0	0
DcTagErr7	0x13	0	0	0	0	0	0
L2DataErr	0x14	0	0	0	0	0	0
ScbStateAdd r	0x15	1	1	1	0	0	0
ScbData0	0x16	0	0	0	1	1	0
ScbData1	0x17	0	0	0	1	1	0
ScbDataNon Cacheable	0x18	0	0	0	1	1	0
SystemRead DataErrorW cb	0x19	1	1	1	0	0	0/1
Hwa	0x1a	1	1	1	0	0	0
Reserved	0x1b	0	0	0	0	0	0

**MSR0000\_0000...MSRC000\_2002 [LS Machine Check Address Thread 0] (MCA::LS::MCA\_ADDR\_LS)**

Read-write. Reset: Cold,0000\_0000\_0000\_0000h.

MCA::LS::MCA\_ADDR\_LS stores an address and other information associated with the error in MCA::LS::MCA\_STATUS\_LS. The register is only meaningful if MCA::LS::MCA\_STATUS\_LS[Val]=1 and MCA::LS::MCA\_STATUS\_LS[AddrV]=1.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst0\_n[23:0]\_aliasMSRSLLEGACY; MSR0000\_0000

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst0\_n[23:0]\_aliasMSRLEGACY; MSR0000\_0402

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst0\_n[23:0]\_aliasMSR; MSRC000\_2002

Bits	Description
63:0	<b>ErrorAddr.</b> Read-write. Reset: Cold,0000_0000_0000_0000h. Unless otherwise specified by an error, contains the address associated with the error logged in MCA::LS::MCA_STATUS_LS. For physical addresses, the most significant bit is given by Core::X86::Cpuid::LongModeInfo[PhysAddrSize].

Table 38: MCA\_ADDR\_LS

Error Type	Bits	Description
LDQ	[63:0]	Reserved
STQ	[63:0]	Reserved
MAB	[63:0]	Reserved
L1DTLB	[63:48] [47:12] [11:0]	Reserved Virtual Address Reserved
DcTagErr5	[63:0]	Reserved
DcTagErr6	[63:0]	Reserved
DcTagErr1	[63:0]	Reserved

IntErrTyp1	[63:0]	Reserved
IntErrTyp2	[63:0]	Reserved
SystemReadDataErrorUcode	[63:48] [47:6]	Reserved Physical Address
SystemReadDataErrorMab	[63:0]	Reserved
DcTagErr2	[63:0]	Reserved
DcDataErr1	[63:48] [47:6] [5:1]	Reserved Physical Address MCA_STATUS_LS[Poison]=1 ? 5'b0 : Physical Address
DcDataErr2	[63:48] [47:1]	Reserved Physical Address
DcDataErr3	[63:48] [47:1]	Reserved Physical Address
DcTagErr4	[63:0]	Reserved
L2DTLB	[63:48] [47:12] [11:0]	Reserved Virtual Address Reserved
PDC	[63:48] [47:12] [11:0]	Reserved Virtual Address Reserved
DcTagErr3	[63:0]	Reserved
DcTagErr7	[63:0]	Reserved
L2DataErr	[63:0]	Reserved
ScbStateAddr	[63:0]	Reserved
ScbData0	[63:0]	Reserved
ScbData1	[63:0]	Reserved
ScbDataNonCacheable	[63:0]	Reserved
SystemReadDataErrorWcb	[63:0]	Reserved
Hwa	[63:0]	Reserved
Reserved	[63:0]	Reserved

**MSR0000\_0403...MSRC000\_2003 [LS Machine Check Miscellaneous 0 Thread 0] (MCA::LS::MCA\_MISC0\_LS)**

Log miscellaneous information associated with errors.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst0\_n[23:0]\_aliasMSRLEGACY; MSR0000\_0403

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst0\_n[23:0]\_aliasMSR; MSRC000\_2003

Bits	Description
63	<b>Valid.</b> Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	<b>CntP.</b> Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	<b>Locked.</b> Reset: 0. 1=Writes to this register are ignored. This bit is set by BIOS to indicate that this register is not available for OS use. BIOS should set this bit if ThresholdIntType is set to SMI. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
60	<b>IntP.</b> Reset: 1. 1=ThresholdIntType can be used to generate interrupts. 0=ThresholdIntType and interrupt generation are not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::LS::MCA_MISC0_LS[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	<b>LvtOffset.</b> Reset: 0h. One per die. For error thresholding interrupts, specifies the address of the LVT entry in the APIC registers as follows: LVT address = (LvtOffset shifted left 4 bits) + 500h (see Core::X86::Apic::ExtendedInterruptLvtEntries). AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::LS::MCA_MISC0_LS[Locked]) ? Read-write : Read-only.
51	<b>CntEn.</b> Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::LS::MCA_MISC0_LS[Locked]) ? Read-write : Read-only.
50:49	<b>ThresholdIntType.</b> Reset: Cold,0h. Specifies the type of interrupt signaled when Ovrlw is set and IntP==1. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[ThresholdLvtOffset]). 10b = SMI trigger event. 11b = Reserved. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::LS::MCA_MISC0_LS[Locked]) ? Read-write : Read-only.
48	<b>Ovrlw.</b> Reset: Cold,0. Set by hardware when ErrCnt transitions from FFEh to FFFh. When this field is set, ErrCnt no longer increments. When this bit is set, the interrupt selected by the ThresholdIntType field is generated. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::LS::MCA_MISC0_LS[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	<b>ErrCnt.</b> Reset: Cold,000h. This is written by software to set the starting value of the error counter. This is incremented by hardware when errors are logged. When this counter overflows, it stays at FFFh (no rollover). The threshold value, written by software, is (FFFh - the desired error count (the number of errors necessary in order for an interrupt to be taken)); the desired error count of 0 (a write value of FFFh) is not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::LS::MCA_MISC0_LS[Locked]) ? Read-write : Read-only.
31:24	<b>BlkPtr.</b> Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

**MSRC000\_2004 [LS Machine Check Configuration Thread 0] (MCA::LS::MCA\_CONFIG\_LS)**

Reset: 0000\_0000\_0000\_0125h.

Controls configuration of the associated machine check bank.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst0\_n[23:0]\_aliasMSR; MSRC000\_2004

Bits	Description
63:41	Reserved.
40	<b>IntEn.</b> Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	<b>DeferredIntType.</b> Read-write. Reset: 0h. Specifies the type of interrupt signaled when a deferred error is logged. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[DeferredLvtOffset]). 10b = SMI trigger event. 11b = Reserved.
36:35	Reserved.
34	<b>LogDeferredInMcaStat.</b> Read-write. Reset: 0. Init: BIOS,1. 1=Log deferred errors in MCA::LS::MCA_STATUS_LS and MCA::LS::MCA_ADDR_LS in addition to MCA::LS::MCA_DESTAT_LS and MCA::LS::MCA_DEADDR_LS. 0=Only log deferred errors in MCA::LS::MCA_DESTAT_LS and MCA::LS::MCA_DEADDR_LS. This bit does not affect logging of deferred errors in MCA::LS::MCA_SYND_LS, MCA::LS::MCA_MISC0_LS.
33	Reserved.
32	<b>McaXEnable.</b> Read-write. Reset: 0. Init: BIOS,1. Check: 1. 1=Software has acknowledged support for the MCAX feature set. 0=Software has not acknowledged support for the MCAX feature set. All uncorrected and fatal errors will cause an ErrorEvent packet to be generated. Deferred error interrupts are configured via Core::X86::Msr::McaIntrCfg.
31:11	Reserved.
10	<b>IntPresent.</b> Read-only. Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::LS::MCA_CONFIG_LS[IntEn].
9	<b>McaFruTextInMca.</b> Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	<b>McaLsbInStatusSupported.</b> Read-only. Reset: 1. 1=MCA::LS::MCA_CONFIG_LS[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	<b>DeferredIntTypeSupported.</b> Read-only. Reset: 1. 1=MCA::LS::MCA_CONFIG_LS[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::LS::MCA_CONFIG_LS[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	<b>DeferredErrorLoggingSupported.</b> Read-only. Reset: 1. 1=Deferred errors are supported in this MCA bank, and MCA::LS::MCA_CONFIG_LS[LogDeferredInMcaStat] controls the logging behavior of these errors. MCA::LS::MCA_DESTAT_LS and MCA::LS::MCA_DEADDR_LS are supported in this MCA bank. 0=Deferred errors are not supported in this bank.
1	Reserved.
0	<b>McaX.</b> Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::LS::MCA_MISC0_LS[BlkPtr] indicates the presence of the additional MISC registers, but is not used to determine their MSR numbers. Deferred error interrupt type is specifiable by MCA bank. MCA::LS::MCA_STATUS_LS[TCC] is present.

**MSRC000\_2005 [LS IP Identification Thread 0] (MCA::LS::MCA\_IPID\_LS)**

Reset: 0000\_00B0\_0000\_0000h.

The MCA::LS::MCA\_IPID\_LS register is used by software to determine what IP type and revision is associated with the MCA bank.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst0\_n[23:0]\_aliasMSR; MSRC000\_2005

Bits	Description
63:48	<b>McaType</b> . Read-only. Reset: 0000h. The McaType of the MCA bank within this IP.
47:44	<b>InstanceIdHi</b> . Read-write. Reset: 0h. The HI value instance ID of this IP. This is initialized to a unique ID per instance of this register.
43:32	<b>HardwareID</b> . Read-only. Reset: 0B0h. The Hardware ID of the IP associated with this MCA bank.
31:0	<b>InstanceId</b> . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _lthree0_core0_thread0_inst0_n0_aliasMSR: PSP,2080_A200h
	Init: _lthree0_core0_thread1_inst0_n1_aliasMSR: PSP,2080_A300h
	Init: _lthree0_core1_thread0_inst0_n2_aliasMSR: PSP,2082_A200h
	Init: _lthree0_core1_thread1_inst0_n3_aliasMSR: PSP,2082_A300h
	Init: _lthree0_core2_thread0_inst0_n4_aliasMSR: PSP,2084_A200h
	Init: _lthree0_core2_thread1_inst0_n5_aliasMSR: PSP,2084_A300h
	Init: _lthree0_core3_thread0_inst0_n6_aliasMSR: PSP,2086_A200h
	Init: _lthree0_core3_thread1_inst0_n7_aliasMSR: PSP,2086_A300h
	Init: _lthree1_core0_thread0_inst0_n8_aliasMSR: PSP,20C0_A200h
	Init: _lthree1_core0_thread1_inst0_n9_aliasMSR: PSP,20C0_A300h
	Init: _lthree1_core1_thread0_inst0_n10_aliasMSR: PSP,20C2_A200h
	Init: _lthree1_core1_thread1_inst0_n11_aliasMSR: PSP,20C2_A300h
	Init: _lthree1_core2_thread0_inst0_n12_aliasMSR: PSP,20C4_A200h
	Init: _lthree1_core2_thread1_inst0_n13_aliasMSR: PSP,20C4_A300h
	Init: _lthree1_core3_thread0_inst0_n14_aliasMSR: PSP,20C6_A200h
	Init: _lthree1_core3_thread1_inst0_n15_aliasMSR: PSP,20C6_A300h
	Init: _lthree1_core4_thread0_inst0_n16_aliasMSR: PSP,20C8_A200h
	Init: _lthree1_core4_thread1_inst0_n17_aliasMSR: PSP,20C8_A300h
	Init: _lthree1_core5_thread0_inst0_n18_aliasMSR: PSP,20CA_A200h
	Init: _lthree1_core5_thread1_inst0_n19_aliasMSR: PSP,20CA_A300h
	Init: _lthree1_core6_thread0_inst0_n20_aliasMSR: PSP,20CC_A200h
	Init: _lthree1_core6_thread1_inst0_n21_aliasMSR: PSP,20CC_A300h
	Init: _lthree1_core7_thread0_inst0_n22_aliasMSR: PSP,20CE_A200h
	Init: _lthree1_core7_thread1_inst0_n23_aliasMSR: PSP,20CE_A300h

**MSRC000\_2006 [LS Machine Check Syndrome Thread 0] (MCA::LS::MCA\_SYND\_LS)**

Read-write. Reset: Cold,0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::LS::MCA\_STATUS\_LS Thread 0

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst0\_n[23:0]\_aliasMSR; MSRC000\_2006

Bits	Description
63:32	<b>Syndrom</b> . Read-write. Reset: Cold,0000_0000h. Contains the syndrome, if any, associated with the error logged in MCA::LS::MCA_STATUS_LS. The low-order bit of the syndrome is stored in bit 0, and the syndrome has a length specified by MCA::LS::MCA_SYND_LS[Length]. The Syndrome field is only valid when MCA::LS::MCA_SYND_LS[Length] is not 0.
31:27	Reserved.
26:24	<b>ErrorPriority</b> . Read-write. Reset: Cold,0h. Encodes the priority of the error logged in MCA::LS::MCA_SYND_LS. 3'b000 = No error; 3'b001 = Reserved; 3'b010 = Corrected Error; 3'b011 = Deferred Error; 3'b100 = Uncorrected Error; 3'b101 = Fatal Error; all others reserved.
23:18	<b>Length</b> . Read-write. Reset: Cold,00h. Specifies the length in bits of the syndrome contained in MCA::LS::MCA_SYND_LS[Syndrome]. A value of 0 indicates that there is no valid syndrome in MCA::LS::MCA_SYND_LS. For example, a syndrome length of 9 means that MCA::LS::MCA_SYND_LS[Syndrome] bits [8:0] contains a valid syndrome.
17:0	<b>ErrorInformation</b> . Read-write. Reset: Cold,0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 39 [MCA_SYND_LS].

Table 39: MCA\_SYND\_LS

Error Type	Bits	Description
LDQ	[17:0]	Reserved
STQ	[17:0]	Reserved
MAB	[17:0]	Reserved
L1DTLB	[17:0]	Reserved
DcTagErr5	[17:16] [15:8] [7:0]	2'b11 Index Way
DcTagErr6	[17:16] [15:8] [7:0]	2'b11 Index Way
DcTagErr1	[17:16] [15:8] [7:0]	2'b11 Index Way
IntErrTyp1	[17:11] [10] [9:0]	Reserved Thread ID Reserved
IntErrTyp2	[17:12] [11] [10:1] [0]	Reserved Thread ID Reserved Reserved
SystemReadDataErrorUcode	[17:2] [1:0]	Reserved 2'b00 = Decode Error; 2'b01 = Target Abort; 2'b10 = Transaction Error; 2'b11 = Protection Violation
SystemReadDataErrorMab	[17:12] [11:8] [7:2] [1:0]	Reserved DC way holding the miss address where the error occurred Address [11:6] of error 2'b00 = Decode Error; 2'b01 = Target Abort; 2'b10 = Transaction Error; 2'b11 = Protection Violation
DcTagErr2	[17:16] [15:8] [7:0]	2'b11 Index Way
DcDataErr1	[17:16] [15:8] [7:0]	MCA_STATUS_LS[Poison]=1 ? 2'b00 : 2'b11 Index Way
DcDataErr2	[17:16] [15:8] [7:0]	2'b11 Index Way
DcDataErr3	[17:16] [15:14] [13:8] [7:3] [2:0]	2'b11 Reserved Index Physical Address[5:1] Way
DcTagErr4	[17:16] [15:8] [7:0]	Reserved Index Way
L2DTLB	[17:16] [15] [14:8]	2'b11 Reserved Reserved

	[7:4] [3:0]	Reserved Reserved
PDC	[17:0]	Reserved
DcTagErr3	[17:16] [15:8] [7:0]	2'b11 Index Way
DcTagErr7	[17:16] [15:8] [7:0]	2'b11 Index Way
L2DataErr	[17:0]	Reserved
ScbStateAddr	[17:0]	Reserved
ScbData0	[17:0]	Reserved
ScbData1	[17:0]	Reserved
ScbDataNonCacheable	[17:0]	Reserved
SystemReadDataErrorWcb	[17:2] [1:0]	Reserved 2'b00 = Decode Error; 2'b01 = Target Abort; 2'b10 = Transaction Error; 2'b11 = Protection Violation
Hwa	[17:0]	Reserved
Reserved	[17:0]	Reserved



**MSRC000\_2008 [LS Machine Check Deferred Error Status Thread 0] (MCA::LS::MCA\_DESTAT\_LS)**

Read-write. Reset: Cold,0000\_0000\_0000\_0000h.

Holds status information for the first deferred error seen in this bank.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst0\_n[23:0]\_aliasMSR; MSRC000\_2008

Bits	Description
63	<b>Val.</b> Read-write. Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	<b>Overflow.</b> Read-write. Reset: Cold,0. 1=An error was detected while the valid bit (Val) was set; at least one error was not logged. Overflow is set independently of whether the existing error is overwritten. (See the section on error overwrite priorities.)
61:59	<b>RESERV4.</b> Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	<b>AddrV.</b> Read-write. Reset: Cold,0. 1=MCA::LS::MCA_DEADDR_LS contains address information associated with the error.
57:54	<b>RESERV3.</b> Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	<b>SyndV.</b> Read-write. Reset: Cold,0. 1=This error logged information in MCA::LS::MCA_SYND_LS. If MCA::LS::MCA_SYND_LS[ErrorPriority] is the same as the priority of the error in MCA::LS::MCA_STATUS_LS, then the information in MCA::LS::MCA_SYND_LS is associated with the error in MCA::LS::MCA_DESTAT_LS.
52:45	<b>RESERV2.</b> Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	<b>Deferred.</b> Read-write. Reset: Cold,0. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; poison is created and an exception is deferred until the poison data is consumed.
43:30	<b>RESERV1.</b> Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	<b>AddrLsb.</b> Read-write. Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::LS::MCA_ADDR_LS[ErrorAddr]. A value of 0 indicates that MCA::LS::MCA_ADDR_LS[63:0] contains a valid byte address. A value of 6 indicates that MCA::LS::MCA_ADDR_LS[63:6] contains a valid cache line address and that MCA::LS::MCA_ADDR_LS[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::LS::MCA_ADDR_LS[63:12] contain a valid 4KB memory page and that MCA::LS::MCA_ADDR_LS[11:0] should be ignored by error handling software.
23:22	<b>RESERV0.</b> Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	<b>ErrorCodeExt.</b> Read-write. Reset: Cold,00h. Logs an extended error code when an error is detected. This model-specific field is used in conjunction with ErrorCode to identify the error sub-type for root cause analysis.
15:0	<b>ErrorCode.</b> Read-write. Reset: Cold,0000h. Error code for this error.

**MSRC000\_2009 [LS Deferred Error Address Thread 0] (MCA::LS::MCA\_DEADDR\_LS)**

Read-write. Reset: Cold,0000\_0000\_0000\_0000h.

The MCA::LS::MCA\_DEADDR\_LS register stores the address associated with the error in MCA::LS::MCA\_DESTAT\_LS. The register is only meaningful if MCA::LS::MCA\_DESTAT\_LS[Val]=1 and MCA::LS::MCA\_DESTAT\_LS[AddrV]=1. The lowest valid bit of the address is defined by MCA::LS::MCA\_DESTAT\_LS[AddrLsb].

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst0\_n[23:0]\_aliasMSR; MSRC000\_2009

Bits	Description
63:0	<b>ErrorAddr.</b> Read-write. Reset: Cold,0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::LS::MCA_DESTAT_LS. The lowest-order valid bit of the address is specified in MCA::LS::MCA_DESTAT_LS[AddrLsb].

**MSRC001\_0400 [LS Machine Check Control Mask Thread 0] (MCA::LS::MCA\_CTL\_MASK\_LS)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

Inhibit detection of an error source.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst0\_n[23:0]\_aliasMSR; MSRC001\_0400

Bits	Description
63:27	Reserved.
26	<b>Hwa.</b> Read-write. Reset: 0. Hardware Asserts (HWAs)
25	<b>SystemReadDataErrorWcb.</b> Read-write. Reset: 0. Init: BIOS,1. System Read Data Error detected by write combine buffer. An error in a read of a line from the data fabric. Possible reasons include decode error and target abort.
24	<b>ScbDataNonCacheable.</b> Read-write. Reset: 0. Error on SCB data for non-cacheable DRAM or IO, discovered at data-pull time
23	<b>ScbData1.</b> Read-write. Reset: 0. Error on SCB data, commit pipe 1, discovered at SCB commit time
22	<b>ScbData0.</b> Read-write. Reset: 0. Error on SCB data, commit pipe 0, discovered at SCB commit time
21	<b>ScbStateAddr.</b> Read-write. Reset: 0. Error on SCB cacheline state (way and moesi state) or address field
20	<b>L2DataErr.</b> Read-write. Reset: 0. L2 Fill Data error.
19	<b>DcTagErr7.</b> Read-write. Reset: 0. DC Tag error type 5.
18	<b>DcTagErr3.</b> Read-write. Reset: 0. DC Tag error type 3.
17	<b>PDC.</b> Read-write. Reset: 0. PDC parity error.
16	<b>L2DTLB.</b> Read-write. Reset: 0. Level 2 TLB parity error.
15	<b>DcTagErr4.</b> Read-write. Reset: 0. DC Tag error type 4.
14	<b>DcDataErr3.</b> Read-write. Reset: 0. DC Data error type 3.
13	<b>DcDataErr2.</b> Read-write. Reset: 0. DC Data error type 2.
12	<b>DcDataErr1.</b> Read-write. Reset: 0. DC Data error type 1 and poison consumption. MCA_STATUS[Poison] is set on poison consumption from L2/L3.
11	<b>DcTagErr2.</b> Read-write. Reset: 0. DC Tag error type 2.
10	<b>SystemReadDataErrorMab.</b> Read-write. Reset: 0. Init: BIOS,1. System Read Data Error detected by mab. An error in a read of a line from the data fabric. Possible reasons include decode error and target abort.
9	<b>SystemReadDataErrorUcode.</b> Read-write. Reset: 0. Init: BIOS,1. System Read Data Error logged by ucode. An error in a read of a line from the data fabric. Possible reasons include decode error and target abort.
8	<b>IntErrTyp2.</b> Read-write. Reset: 0. Internal error type 2.
7	<b>IntErrTyp1.</b> Read-write. Reset: 0. Internal error type 1.
6	<b>DcTagErr1.</b> Read-write. Reset: 0. DC Tag error type 1.
5	<b>DcTagErr6.</b> Read-write. Reset: 0. DC Tag error type 6.
4	<b>DcTagErr5.</b> Read-write. Reset: 0. DC Tag error type 5.
3	<b>L1DTLB.</b> Read-write. Reset: 0. Level 1 TLB parity error.
2	<b>MAB.</b> Read-write. Reset: 0. Miss address buffer payload parity error.
1	<b>STQ.</b> Read-write. Reset: 0. Store queue parity error.
0	<b>LDQ.</b> Read-write. Reset: 0. Load queue parity error.

**MSRC000\_200E [LS Machine Check Syndrome Extended Thread 0] (MCA::LS::MCA\_SYND1\_LS)**

Read-write. Reset: Cold,0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::LS::MCA\_STATUS\_LS Thread 0

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst0\_n[23:0]\_aliasMSR; MSRC000\_200E

Bits	Description
63:0	<b>Syndrome.</b> Read-write. Reset: Cold,0000_0000_0000_0000h. The MCA::LS::MCA_SYND1_LS register stores information associated with the error in MCA::LS::MCA_STATUS_LS or MCA_DESTAT. The register is meaningful if MCA::LS::MCA_STATUS_LS[SyndV]=1. When MCA::LS::MCA_CONFIG_LS[McaFruTextInMca]=1, MCA::LS::MCA_SYND1_LS stores ASCII FruText associated with the error.

**MSRC000\_200F [LS Machine Check Syndrome Extended Thread 0] (MCA::LS::MCA\_SYND2\_LS)**

Read-write. Reset: Cold,0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::LS::MCA\_STATUS\_LS Thread 0

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst0\_n[23:0]\_aliasMSR; MSRC000\_200F

Bits	Description
63:0	<b>Syndrone.</b> Read-write. Reset: Cold,0000_0000_0000_0000h. The MCA::LS::MCA_SYND2_LS register stores information associated with the error in MCA::LS::MCA_STATUS_LS or MCA_DESTAT. The register is meaningful if MCA::LS::MCA_STATUS_LS[SyndV]=1. When MCA::LS::MCA_CONFIG_LS[McaFruTextInMca]=1, MCA::LS::MCA_SYND2_LS stores ASCII FruText associated with the error.

**3.2.5.2 IF****MSR0000\_0404...MSRC000\_2010 [IF Machine Check Control Thread 0] (MCA::IF::MCA\_CTL\_IF)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

0=Disables error reporting for the corresponding error. 1=Enables error reporting via machine check exception for the corresponding error. The MCA::IF::MCA\_CTL\_IF register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG\_CTL. Does not affect error detection, correction, or logging.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst1\_n[23:0]\_aliasMSRLEGACY; MSR0000\_0404

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst1\_n[23:0]\_aliasMSR; MSRC000\_2010

Bits	Description
63:20	Reserved.
19	<b>RSVD19.</b> Read-write. Reset: 0.
18	<b>CtMceError.</b> Read-write. Reset: 0. CT MCE
17	<b>RSVD17.</b> Read-write. Reset: 0. Reserved. Will never trigger.
16	<b>L2TlbMultiHit.</b> Read-write. Reset: 0. L2-TLB Multi-Hit
15	<b>L1TlbMultiHit.</b> Read-write. Reset: 0. L1-TLB Multi-Hit.
14	<b>HwAssert.</b> Read-write. Reset: 0. Hardware Assertion Error.
13	<b>SystemReadDataError.</b> Read-write. Reset: 0. System Read Data Error. An error in a demand fetch of a line. Possible reasons include decode error and target abort.
12	<b>L2RespPoison.</b> Read-write. Reset: 0. L2 Cache Response Poison Error. Error is the result of consuming poison data.
11	<b>L2BtbMultiHit.</b> Read-write. Reset: 0. L2 BTB Multi-Match Error.
10	<b>L1BtbMultiHit.</b> Read-write. Reset: 0. L1 BTB Multi-Match Error.
9	<b>BpqSnpParT1.</b> Read-write. Reset: 0. BPQ Thread 1 Snoop Parity Error.
8	<b>BpqSnpParT0.</b> Read-write. Reset: 0. BPQ Thread 0 Snoop Parity Error.
7	<b>L2ItlbParity.</b> Read-write. Reset: 0. L2 ITLB Parity Error.
6	<b>L1ItlbParity.</b> Read-write. Reset: 0. L1 ITLB Parity Error.
5	<b>RSVD5.</b> Read-write. Reset: 0. Reserved. Will never trigger.
4	<b>DqParity.</b> Read-write. Reset: 0. Decoupling Queue PhysAddr Parity Error.
3	<b>DataParity.</b> Read-write. Reset: 0. IC Data Array Parity Error.
2	<b>TagParity.</b> Read-write. Reset: 0. IC Full Tag Parity Error.
1	<b>TagMultiHit.</b> Read-write. Reset: 0. IC Microtag or Full Tag Multi-hit Error.
0	<b>OcUtagParity.</b> Read-write. Reset: 0. Op Cache Microtag Probe Port Parity Error.

**MSR0000\_0405...MSRC000\_2011 [IF Machine Check Status Thread 0] (MCA::IF::MCA\_STATUS\_IF)**

Reset: Cold,0000\_0000\_0000\_0000h.

Logs information associated with errors.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst1\_n[23:0]\_aliasMSRLEGACY; MSR0000\_0405

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst1\_n[23:0]\_aliasMSR; MSRC000\_2011

Bits	Description
63	<b>Val.</b> Reset: Cold,0. 1=A valid error has been detected. This bit should be cleared by software after the register has been read. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
62	<b>Overflow.</b> Reset: Cold,0. 1=An error was detected while the valid bit (Val) was set; at least one error was not logged. Overflow is set independently of whether the existing error is overwritten. See 3.1.3 [Machine Check Errors]. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
61	<b>UC.</b> Reset: Cold,0. 1=The error was not corrected by hardware. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
60	<b>En.</b> Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::IF::MCA_CTL_IF. This bit is a copy of bit in MCA::IF::MCA_CTL_IF for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	<b>MiscV.</b> Reset: Cold,0. 1=Valid thresholding in MCA::IF::MCA_MISC0_IF. In certain modes, MISC registers are owned by platform firmware and will RAZ when read by non-SMM code. Therefore, it is possible for MiscV=1 and the MISC register to read as all zeros. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
58	<b>AddrV.</b> Reset: Cold,0. 1=MCA::IF::MCA_ADDR_IF contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	<b>PCC.</b> Reset: Cold,0. 1=Hardware context held by the processor may have been corrupted. Continued operation of the system may have unpredictable results. The error is not recoverable or survivable, and the system should be reinitialized. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
56	<b>ErrCoreIdVal.</b> Reset: Cold,0. 1=The ErrCoreId field is valid. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
55	<b>TCC.</b> Reset: Cold,0. 1=Hardware context of the process thread to which the error was reported may have been corrupted. Continued operation of the thread may have unpredictable results. The thread must be terminated. Only meaningful when MCA::IF::MCA_STATUS_IF[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	<b>RESERV54.</b> Reset: Cold,0. MCA_STATUS Register Reserved bit. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
53	<b>SyndV.</b> Reset: Cold,0. 1=This error logged information in MCA::IF::MCA_SYND_IF. If MCA::IF::MCA_SYND_IF[ErrorPriority] is the same as the priority of the error in MCA::IF::MCA_STATUS_IF, then the information in MCA::IF::MCA_SYND_IF is associated with the error in MCA::IF::MCA_STATUS_IF. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	<b>RESERV47.</b> Reset: Cold,00h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
46	<b>CECC.</b> Reset: Cold,0. 1=The error was a correctable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
45	<b>UECC.</b> Reset: Cold,0. 1=The error was an uncorrectable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.

44	<b>Deferred.</b> Reset: Cold,0. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; an exception is deferred until the erroneous data is consumed. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
43	<b>Poison.</b> Reset: Cold,0. 1=The error was the result of attempting to consume poisoned data. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
42:41	<b>RESERV41.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
40	<b>Scrub.</b> Reset: Cold,0. 1=The error was the result of a scrub operation. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
39:38	<b>RESERV38.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
37:32	<b>ErrCoreId.</b> Reset: Cold,00h. When ErrCoreIdVal=1 this field indicates which core within the processor is associated with the error; Otherwise this field is reserved. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
31:30	<b>RESERV30.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
29:24	<b>AddrLsb.</b> Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::IF::MCA_ADDR_IF[ErrorAddr]. A value of 0 indicates that MCA::IF::MCA_ADDR_IF[63:0] contains a valid byte address. A value of 6 indicates that MCA::IF::MCA_ADDR_IF[63:6] contains a valid cache line address and that MCA::IF::MCA_ADDR_IF[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::IF::MCA_ADDR_IF[63:12] contain a valid 4KB memory page and that MCA::IF::MCA_ADDR_IF[11:0] should be ignored by error handling software. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
23:22	<b>RESERV22.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
21:16	<b>ErrorCodeExt.</b> Reset: Cold,00h. Extended Error Code. This field is used to identify the error type for root cause analysis. This field indicates which bit position in MCA::IF::MCA_CTL_IF enables error reporting for the logged error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
15:0	<b>ErrorCode.</b> Reset: Cold,0000h. Error code for this error. See 3.1.3.3 [Error Codes] for details on decoding this field. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.

Table 40: MCA\_STATUS\_IF

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
OcUtagParity	0x0	1	1	1	0	0	0
TagMultiHit	0x1	0	0	0	0	0	1
TagParity	0x2	0	0	0	0	0	1
DataParity	0x3	0	0	0	0	0	1
DqParity	0x4	1	1	1	0	0	1
RSVD5	0x5	1	1	1	0	0	1
L1ItlbParity	0x6	1	1	1	0	0	1
L2ItlbParity	0x7	0	0	0	0	0	1
BpqSnpParT0	0x8	0	0	0	0	0	0
BpqSnpParT	0x9	0	0	0	0	0	0

1							
L1BtbMulti Hit	0xa	0	0	0	0	0	0
L2BtbMulti Hit	0xb	0	0	0	0	0	0
L2RespPoison	0xc	1	0	1	0	1	1
SystemReadDataError	0xd	1	0	1	0	0	1
HwAssert	0xe	1	1	1	0	0	0
L1TlbMulti Hit	0xf	1	1	1	0	0	1
L2TlbMulti Hit	0x10	0	0	0	0	0	1
RSVD17	0x11	1	1	1	0	0	0
CtMceError	0x12	1	1	1	0	0	0
RSVD19	0x13	0	0	0	0	0	1

**MSR0000\_0406...MSRC000\_2012 [IF Machine Check Address Thread 0] (MCA::IF::MCA\_ADDR\_IF)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

MCA::IF::MCA\_ADDR\_IF stores an address and other information associated with the error in MCA::IF::MCA\_STATUS\_IF. The register is only meaningful if MCA::IF::MCA\_STATUS\_IF[Val]=1 and MCA::IF::MCA\_STATUS\_IF[AddrV]=1.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst1\_n[23:0]\_aliasMSRLEGACY; MSR0000\_0406

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst1\_n[23:0]\_aliasMSR; MSRC000\_2012

Bits	Description
63:0	<b>ErrorAddr.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. Unless otherwise specified by an error, contains the address associated with the error logged in MCA::IF::MCA_STATUS_IF. For physical addresses, the most significant bit is given by Core::X86::Cpuid::LongModeInfo[PhysAddrSize].

Table 41: MCA\_ADDR\_IF

Error Type	Bits	Description
OcUtagParity	[63:0]	Reserved
TagMultiHit	[63:52] [51:0]	Reserved Physical Address
TagParity	[63:52] [51:0]	Reserved Physical Address
DataParity	[63:52] [51:0]	Reserved Physical Address
DqParity	[63:52] [51:0]	Reserved Physical Address
RSVD5	[63:57] [56:12] [11:0]	Reserved Linear Address Reserved
L1ItlbParity	[63:57] [56:12] [11:0]	Reserved Linear Address Reserved
L2ItlbParity	[63:57] [56:12] [11:0]	Reserved Linear Address Reserved



BpqSnpParT0	[63:0]	Reserved
BpqSnpParT1	[63:0]	Reserved
L1BtbMultiHit	[63:0]	Reserved
L2BtbMultiHit	[63:0]	Reserved
L2RespPoison	[63:52] [51:5] [4:0]	Reserved Physical Address Reserved
SystemReadDataError	[63:52] [51:5] [4:0]	Reserved Physical Address Reserved
HwAssert	[56:0]	Reserved
L1TlbMultiHit	[56:0]	VA
L2TlbMultiHit	[56:0]	VA
RSVD17	[56:0]	Reserved
CtMceError	[56:0]	Reserved
RSVD19	[63:0]	Reserved

**MSR0000\_0407...MSRC000\_2013 [IF Machine Check Miscellaneous 0 Thread 0] (MCA::IF::MCA\_MISC0\_IF)**

Log miscellaneous information associated with errors.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst1\_n[23:0]\_aliasMSRLEGACY; MSR0000\_0407

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst1\_n[23:0]\_aliasMSR; MSRC000\_2013

Bits	Description
63	<b>Valid.</b> Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	<b>CntP.</b> Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	<b>Locked.</b> Reset: 0. 1=Writes to this register are ignored. This bit is set by BIOS to indicate that this register is not available for OS use. BIOS should set this bit if ThresholdIntType is set to SMI. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
60	<b>IntP.</b> Reset: 1. 1=ThresholdIntType can be used to generate interrupts. 0=ThresholdIntType and interrupt generation are not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::IF::MCA_MISC0_IF[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	<b>LvtOffset.</b> Reset: 0h. One per die. For error thresholding interrupts, specifies the address of the LVT entry in the APIC registers as follows: LVT address = (LvtOffset shifted left 4 bits) + 500h (see Core::X86::Apic::ExtendedInterruptLvtEntries). AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::IF::MCA_MISC0_IF[Locked]) ? Read-write : Read-only.
51	<b>CntEn.</b> Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::IF::MCA_MISC0_IF[Locked]) ? Read-write : Read-only.
50:49	<b>ThresholdIntType.</b> Reset: Cold,0h. Specifies the type of interrupt signaled when Ovrlw is set and IntP==1. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[ThresholdLvtOffset]). 10b = SMI trigger event. 11b = Reserved. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::IF::MCA_MISC0_IF[Locked]) ? Read-write : Read-only.
48	<b>Ovrlw.</b> Reset: Cold,0. Set by hardware when ErrCnt transitions from FFEh to FFFh. When this field is set, ErrCnt no longer increments. When this bit is set, the interrupt selected by the ThresholdIntType field is generated. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::IF::MCA_MISC0_IF[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	<b>ErrCnt.</b> Reset: Cold,000h. This is written by software to set the starting value of the error counter. This is incremented by hardware when errors are logged. When this counter overflows, it stays at FFFh (no rollover). The threshold value, written by software, is (FFFh - the desired error count (the number of errors necessary in order for an interrupt to be taken)); the desired error count of 0 (a write value of FFFh) is not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::IF::MCA_MISC0_IF[Locked]) ? Read-write : Read-only.
31:24	<b>BlkPtr.</b> Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.



**MSRC000\_2014 [IF Machine Check Configuration Thread 0] (MCA::IF::MCA\_CONFIG\_IF)**

Reset: 0000\_0000\_0000\_0121h.

Controls configuration of the associated machine check bank.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst1\_n[23:0]\_aliasMSR; MSRC000\_2014

Bits	Description
63:41	Reserved.
40	<b>IntEn.</b> Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	<b>DeferredIntType.</b> Read-write. Reset: 0h. Specifies the type of interrupt signaled when a deferred error is logged. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[DeferredLvtOffset]). 10b = SMI trigger event. 11b = Reserved.
36:33	Reserved.
32	<b>McaXEnable.</b> Read-write. Reset: 0. Init: BIOS,1. Check: 1. 1=Software has acknowledged support for the MCAX feature set. 0=Software has not acknowledged support for the MCAX feature set. All uncorrected and fatal errors will cause an ErrorEvent packet to be generated. Deferred error interrupts are configured via Core::X86::Msr::McaIntrCfg.
31:11	Reserved.
10	<b>IntPresent.</b> Read-only, Volatile. Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::IF::MCA_CONFIG_IF[IntEn].
9	<b>McaFruTextInMca.</b> Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	<b>McaLsbInStatusSupported.</b> Read-only. Reset: 1. 1=MCA::IF::MCA_CONFIG_IF[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	<b>DeferredIntTypeSupported.</b> Read-only. Reset: 1. 1=MCA::IF::MCA_CONFIG_IF[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::IF::MCA_CONFIG_IF[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	<b>DeferredErrorLoggingSupported.</b> Read-only. Reset: 0. 1=Deferred errors are supported in this MCA bank, and the LogDeferredInMcaStat field in this register controls the logging behavior of these errors. 0=Deferred errors are not supported in this bank.
1	Reserved.
0	<b>McaX.</b> Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::IF::MCA_MISC0_IF[BlkPtr] indicates the presence of the additional MISC registers, but is not used to determine their MSR numbers. Deferred error interrupt type is specifiable by MCA bank. MCA::IF::MCA_STATUS_IF[TCC] is present.

**MSRC000\_2015 [IF IP Identification Thread 0] (MCA::IF::MCA\_IPID\_IF)**

Reset: 0001\_00B0\_0000\_0000h.

The MCA::IF::MCA\_IPID\_IF register is used by software to determine what IP type and revision is associated with the MCA bank.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst1\_n[23:0]\_aliasMSR; MSRC000\_2015

Bits	Description
63:48	<b>McaType.</b> Read-only. Reset: 0001h. The McaType of the MCA bank within this IP.
47:44	<b>InstanceIdHi.</b> Read-write. Reset: 0h. The HI value instance ID of this IP. This is initialized to a unique ID per instance of this register.
43:32	<b>HardwareID.</b> Read-only. Reset: 0B0h. The Hardware ID of the IP associated with this MCA bank.
31:0	<b>InstanceId.</b> Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _lthree0_core0_thread0_inst1_n0_aliasMSR: PSP,2080_AA00h
	Init: _lthree0_core0_thread1_inst1_n1_aliasMSR: PSP,2080_AB00h
	Init: _lthree0_core1_thread0_inst1_n2_aliasMSR: PSP,2082_AA00h
	Init: _lthree0_core1_thread1_inst1_n3_aliasMSR: PSP,2082_AB00h
	Init: _lthree0_core2_thread0_inst1_n4_aliasMSR: PSP,2084_AA00h
	Init: _lthree0_core2_thread1_inst1_n5_aliasMSR: PSP,2084_AB00h
	Init: _lthree0_core3_thread0_inst1_n6_aliasMSR: PSP,2086_AA00h
	Init: _lthree0_core3_thread1_inst1_n7_aliasMSR: PSP,2086_AB00h
	Init: _lthree1_core0_thread0_inst1_n8_aliasMSR: PSP,20C0_AA00h
	Init: _lthree1_core0_thread1_inst1_n9_aliasMSR: PSP,20C0_AB00h
	Init: _lthree1_core1_thread0_inst1_n10_aliasMSR: PSP,20C2_AA00h
	Init: _lthree1_core1_thread1_inst1_n11_aliasMSR: PSP,20C2_AB00h
	Init: _lthree1_core2_thread0_inst1_n12_aliasMSR: PSP,20C4_AA00h
	Init: _lthree1_core2_thread1_inst1_n13_aliasMSR: PSP,20C4_AB00h
	Init: _lthree1_core3_thread0_inst1_n14_aliasMSR: PSP,20C6_AA00h
	Init: _lthree1_core3_thread1_inst1_n15_aliasMSR: PSP,20C6_AB00h
	Init: _lthree1_core4_thread0_inst1_n16_aliasMSR: PSP,20C8_AA00h
	Init: _lthree1_core4_thread1_inst1_n17_aliasMSR: PSP,20C8_AB00h
	Init: _lthree1_core5_thread0_inst1_n18_aliasMSR: PSP,20CA_AA00h
	Init: _lthree1_core5_thread1_inst1_n19_aliasMSR: PSP,20CA_AB00h
	Init: _lthree1_core6_thread0_inst1_n20_aliasMSR: PSP,20CC_AA00h
	Init: _lthree1_core6_thread1_inst1_n21_aliasMSR: PSP,20CC_AB00h
	Init: _lthree1_core7_thread0_inst1_n22_aliasMSR: PSP,20CE_AA00h
	Init: _lthree1_core7_thread1_inst1_n23_aliasMSR: PSP,20CE_AB00h

**MSRC000\_2016 [IF Machine Check Syndrome Thread 0] (MCA::IF::MCA\_SYND\_IF)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::IF::MCA\_STATUS\_IF Thread 0

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst1\_n[23:0]\_aliasMSR; MSRC000\_2016

Bits	Description
63:32	<b>Syndrome.</b> Read-write, Volatile. Reset: Cold, 0000_0000h. Contains the syndrome, if any, associated with the error logged in MCA::IF::MCA_STATUS_IF. The low-order bit of the syndrome is stored in bit 0, and the syndrome has a length specified by MCA::IF::MCA_SYND_IF[Length]. The Syndrome field is only valid when MCA::IF::MCA_SYND_IF[Length] is not 0.
31:27	Reserved.
26:24	<b>ErrorPriority.</b> Read-write, Volatile. Reset: Cold, 0h. Encodes the priority of the error logged in MCA::IF::MCA_SYND_IF. 3'b000 = No error; 3'b001 = Reserved; 3'b010 = Corrected Error; 3'b011 = Deferred Error; 3'b100 = Uncorrected Error; 3'b101 = Fatal Error; all others reserved.
23:18	<b>Length.</b> Read-write, Volatile. Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::IF::MCA_SYND_IF[Syndrome]. A value of 0 indicates that there is no valid syndrome in MCA::IF::MCA_SYND_IF. For example, a syndrome length of 9 means that MCA::IF::MCA_SYND_IF[Syndrome] bits [8:0] contains a valid syndrome.
17:0	<b>ErrorInformation.</b> Read-write, Volatile. Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 42 [MCA_SYND_IF].

Table 42: MCA\_SYND\_IF

Error Type	Bits	Description
OcUtagParity	[17:6] [5:0]	Reserved Index
TagMultiHit	[17:16] [15:8] [8:0]	Reserved Subcache Reserved
TagParity	[17:8] [7:0]	Reserved Way
DataParity	[17:16] [15:8] [8:0]	Reserved Subcache Way
DqParity	[17:0]	Reserved
RSVD5	[17:4] [3:0]	Reserved Reserved
L1ItlbParity	[17:6] [5:0]	Reserved Reserved
L2ItlbParity	[17:8] [7:0]	Reserved Reserved
BpqSnpParT0	[17:0]	Reserved
BpqSnpParT1	[17:6] [5:0]	Reserved Index
L1BtbMultiHit	[17:0]	Reserved
L2BtbMultiHit	[17:0]	Reserved
L2RespPoison	[17:0]	Reserved
SystemReadDataError	[17:2] [1:0]	Reserved 2'b00 = Decode Error; 2'b01 = Target Abort; 2'b10 = Transaction Error; 2'b11 = Protection Violation
HwAssert	[17:0]	HwaMcaCode
L1TlbMultiHit	[17:6] [5:0]	Reserved Index
L2TlbMultiHit	[17:8] [7:0]	Reserved Index
RSVD17	[17:0]	Reserved
CtMceError	[17:2] [1:0]	Reserved Thread bit vector
RSVD19	[17:0]	Reserved

**MSRC001\_0401 [IF Machine Check Control Mask Thread 0] (MCA::IF::MCA\_CTL\_MASK\_IF)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

Inhibit detection of an error source.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst1\_n[23:0]\_aliasMSR; MSRC001\_0401

Bits	Description
63:20	Reserved.
19	<b>RSVD19</b> . Read-write. Reset: 0.
18	<b>CtMceError</b> . Read-write. Reset: 0. CT MCE
17	<b>RSVD17</b> . Read-write. Reset: 0. Reserved. Will never trigger.
16	<b>L2TlbMultiHit</b> . Read-write. Reset: 0. L2-TLB Multi-Hit
15	<b>L1TlbMultiHit</b> . Read-write. Reset: 0. L1-TLB Multi-Hit.
14	<b>HwAssert</b> . Read-write. Reset: 0. Hardware Assertion Error.
13	<b>SystemReadDataError</b> . Read-write. Reset: 0. System Read Data Error. An error in a demand fetch of a line. Possible reasons include decode error and target abort.
12	<b>L2RespPoison</b> . Read-write. Reset: 0. L2 Cache Response Poison Error. Error is the result of consuming poison data.
11	<b>L2BtbMultiHit</b> . Read-write. Reset: 0. L2 BTB Multi-Match Error.
10	<b>L1BtbMultiHit</b> . Read-write. Reset: 0. L1 BTB Multi-Match Error.
9	<b>BpqSnpParT1</b> . Read-write. Reset: 0. BPQ Thread 1 Snoop Parity Error.
8	<b>BpqSnpParT0</b> . Read-write. Reset: 0. BPQ Thread 0 Snoop Parity Error.
7	<b>L2ItlbParity</b> . Read-write. Reset: 0. L2 ITLB Parity Error.
6	<b>L1ItlbParity</b> . Read-write. Reset: 0. L1 ITLB Parity Error.
5	<b>RSVD5</b> . Read-write. Reset: 0. Reserved. Will never trigger.
4	<b>DqParity</b> . Read-write. Reset: 0. Decoupling Queue PhysAddr Parity Error.
3	<b>DataParity</b> . Read-write. Reset: 0. IC Data Array Parity Error.
2	<b>TagParity</b> . Read-write. Reset: 0. IC Full Tag Parity Error.
1	<b>TagMultiHit</b> . Read-write. Reset: 0. IC Microtag or Full Tag Multi-hit Error.
0	<b>OcUtagParity</b> . Read-write. Reset: 0. Op Cache Microtag Probe Port Parity Error.

**MSRC000\_201E [IF Machine Check Syndrome Extended Thread 0] (MCA::IF::MCA\_SYND1\_IF)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::IF::MCA\_STATUS\_IF Thread 0

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst1\_n[23:0]\_aliasMSR; MSRC000\_201E

Bits	Description
63:0	<b>Syndrome</b> . Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::IF::MCA_SYND1_IF register stores information associated with the error in MCA::IF::MCA_STATUS_IF or MCA_DESTAT. The register is meaningful if MCA::IF::MCA_STATUS_IF[SyndV]=1. When MCA::IF::MCA_CONFIG_IF[McaFruTextInMca]=1, MCA::IF::MCA_SYND1_IF stores ASCII FruText associated with the error.

**MSRC000\_201F [IF Machine Check Syndrome Extended Thread 0] (MCA::IF::MCA\_SYND2\_IF)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::IF::MCA\_STATUS\_IF Thread 0

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst1\_n[23:0]\_aliasMSR; MSRC000\_201F

Bits	Description
63:0	<b>Syndrome</b> . Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::IF::MCA_SYND2_IF register stores information associated with the error in MCA::IF::MCA_STATUS_IF or MCA_DESTAT. The register is meaningful if MCA::IF::MCA_STATUS_IF[SyndV]=1. When MCA::IF::MCA_CONFIG_IF[McaFruTextInMca]=1, MCA::IF::MCA_SYND2_IF stores ASCII FruText associated with the error.

## 3.2.5.3 L2

**MSR0000\_0408...MSRC000\_2020 [L2 Machine Check Control Thread 0] (MCA::L2::MCA\_CTL\_L2)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

0=Disables error reporting for the corresponding error. 1=Enables error reporting via machine check exception for the corresponding error. The MCA::L2::MCA\_CTL\_L2 register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG\_CTL. Does not affect error detection, correction, or logging.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst2\_n[23:0]\_aliasMSRLEGACY; MSR0000\_0408

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst2\_n[23:0]\_aliasMSR; MSRC000\_2020

Bits	Description
63:7	Reserved.
6	<b>Wdt.</b> Read-write. Reset: 0. Reserved
5	<b>StateMachine.</b> Read-write. Reset: 0. Error initiated by programmable state machine.
4	<b>Sdp.</b> Read-write. Reset: 0. SDP Read Response Parity Error
3	<b>Hwa.</b> Read-write. Reset: 0. Hardware Assert Error.
2	<b>Data.</b> Read-write. Reset: 0. L2M Data Array ECC Error.
1	<b>Tag.</b> Read-write. Reset: 0. L2M Tag or State Array ECC Error.
0	<b>MultiHit.</b> Read-write. Reset: 0. L2M Tag Multiple-Way-Hit error.

**MSR0000\_0409...MSRC000\_2021 [L2 Machine Check Status Thread 0] (MCA::L2::MCA\_STATUS\_L2)**

Reset: Cold,0000\_0000\_0000\_0000h.

Logs information associated with errors.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst2\_n[23:0]\_aliasMSRLEGACY; MSR0000\_0409

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst2\_n[23:0]\_aliasMSR; MSRC000\_2021

Bits	Description
63	<b>Val.</b> Reset: Cold,0. 1=A valid error has been detected. This bit should be cleared by software after the register has been read. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
62	<b>Overflow.</b> Reset: Cold,0. 1=An error was detected while the valid bit (Val) was set; at least one error was not logged. Overflow is set independently of whether the existing error is overwritten. See 3.1.3 [Machine Check Errors]. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
61	<b>UC.</b> Reset: Cold,0. 1=The error was not corrected by hardware. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
60	<b>En.</b> Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::L2::MCA_CTL_L2. This bit is a copy of bit in MCA::L2::MCA_CTL_L2 for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	<b>MiscV.</b> Reset: Cold,0. 1=Valid thresholding in MCA::L2::MCA_MISC0_L2. In certain modes, MISC registers are owned by platform firmware and will RAZ when read by non-SMM code. Therefore, it is possible for MiscV=1 and the MISC register to read as all zeros. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
58	<b>AddrV.</b> Reset: Cold,0. 1=MCA::L2::MCA_ADDR_L2 contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	<b>PCC.</b> Reset: Cold,0. 1=Hardware context held by the processor may have been corrupted. Continued operation of the system may have unpredictable results. The error is not recoverable or survivable, and the system should be reinitialized. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
56	<b>ErrCoreIdVal.</b> Reset: Cold,0. 1=The ErrCoreId field is valid. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
55	<b>TCC.</b> Reset: Cold,0. 1=Hardware context of the process thread to which the error was reported may have been corrupted. Continued operation of the thread may have unpredictable results. The thread must be terminated. Only meaningful when MCA::L2::MCA_STATUS_L2[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	<b>RESERV54.</b> Reset: Cold,0. MCA_STATUS Register Reserved bit. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
53	<b>SyndV.</b> Reset: Cold,0. 1=This error logged information in MCA::L2::MCA_SYND_L2. If MCA::L2::MCA_SYND_L2[ErrorPriority] is the same as the priority of the error in MCA::L2::MCA_STATUS_L2, then the information in MCA::L2::MCA_SYND_L2 is associated with the error in MCA::L2::MCA_STATUS_L2. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	<b>RESERV47.</b> Reset: Cold,00h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
46	<b>CECC.</b> Reset: Cold,0. 1=The error was a correctable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
45	<b>UECC.</b> Reset: Cold,0. 1=The error was an uncorrectable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.

44	<b>Deferred.</b> Reset: Cold,0. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; an exception is deferred until the erroneous data is consumed. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
43	<b>Poison.</b> Reset: Cold,0. 1=The error was the result of attempting to consume poisoned data. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
42:41	<b>RESERV41.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
40	<b>Scrub.</b> Reset: Cold,0. 1=The error was the result of a scrub operation. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
39:38	<b>RESERV38.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
37:32	<b>ErrCoreId.</b> Reset: Cold,00h. When ErrCoreIdVal=1 this field indicates which core within the processor is associated with the error; Otherwise this field is reserved. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
31:30	<b>RESERV30.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
29:24	<b>AddrLsb.</b> Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::L2::MCA_ADDR_L2[ErrorAddr]. A value of 0 indicates that MCA::L2::MCA_ADDR_L2[63:0] contains a valid byte address. A value of 6 indicates that MCA::L2::MCA_ADDR_L2[63:6] contains a valid cache line address and that MCA::L2::MCA_ADDR_L2[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::L2::MCA_ADDR_L2[63:12] contain a valid 4KB memory page and that MCA::L2::MCA_ADDR_L2[11:0] should be ignored by error handling software. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
23:22	<b>RESERV22.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
21:16	<b>ErrorCodeExt.</b> Reset: Cold,00h. Extended Error Code. This field is used to identify the error type for root cause analysis. This field indicates which bit position in MCA::L2::MCA_CTL_L2 enables error reporting for the logged error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
15:0	<b>ErrorCode.</b> Reset: Cold,0000h. Error code for this error. See 3.1.3.3 [Error Codes] for details on decoding this field. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.

Table 43: MCA\_STATUS\_L2

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
MultiHit	0x0	1	1	1	0	0	1
Tag	0x1	0/1	0/1	0/1	0	0	1
Data	0x2	0/1	0/1	0/1	0/1	0	1
Hwa	0x3	1	1	1	0	0	1
Sdp	0x4	0/1	0/1	0/1	0/1	0	0
StateMachine	0x5	0/1	0/1	0/1	0/1	0	1
Wdt	0x6	0	0	0	0	0	1



**MSR0000\_040A...MSRC000\_2022 [L2 Machine Check Address Thread 0] (MCA::L2::MCA\_ADDR\_L2)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

MCA::L2::MCA\_ADDR\_L2 stores an address and other information associated with the error in MCA::L2::MCA\_STATUS\_L2. The register is only meaningful if MCA::L2::MCA\_STATUS\_L2[Val]=1 and MCA::L2::MCA\_STATUS\_L2[AddrV]=1.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst2\_n[23:0]\_aliasMSRLEGACY; MSR0000\_040A

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst2\_n[23:0]\_aliasMSR; MSRC000\_2022

Bits	Description
63:0	<b>ErrorAddr.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. Unless otherwise specified by an error, contains the address associated with the error logged in MCA::L2::MCA_STATUS_L2. For physical addresses, the most significant bit is given by Core::X86::Cpuid::LongModeInfo[PhysAddrSize].

Table 44: MCA\_ADDR\_L2

Error Type	Bits	Description
MultiHit	[55:52] [51:6] [5:0]	Reserved Physical Address Reserved
Tag	[55:52] [51:6] [5:0]	Reserved Physical Address Reserved
Data	[55:52] [51:6] [5:0]	Reserved Physical Address Reserved
Hwa	[31:0]	Reserved
Sdp	[55:0]	Reserved
StateMachine	[63:0]	Reserved
Wdt	[63:0]	Reserved



**MSR0000\_040B...MSRC000\_2023 [L2 Machine Check Miscellaneous 0 Thread 0] (MCA::L2::MCA\_MISC0\_L2)**

Log miscellaneous information associated with errors.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst2\_n[23:0]\_aliasMSRLEGACY; MSR0000\_040B

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst2\_n[23:0]\_aliasMSR; MSRC000\_2023

Bits	Description
63	<b>Valid.</b> Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	<b>CntP.</b> Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	<b>Locked.</b> Reset: 0. 1=Writes to this register are ignored. This bit is set by BIOS to indicate that this register is not available for OS use. BIOS should set this bit if ThresholdIntType is set to SMI. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
60	<b>IntP.</b> Reset: 1. 1=ThresholdIntType can be used to generate interrupts. 0=ThresholdIntType and interrupt generation are not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::L2::MCA_MISC0_L2[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	<b>LvtOffset.</b> Reset: 0h. One per die. For error thresholding interrupts, specifies the address of the LVT entry in the APIC registers as follows: LVT address = (LvtOffset shifted left 4 bits) + 500h (see Core::X86::Apic::ExtendedInterruptLvtEntries). AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::L2::MCA_MISC0_L2[Locked]) ? Read-write : Read-only.
51	<b>CntEn.</b> Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::L2::MCA_MISC0_L2[Locked]) ? Read-write : Read-only.
50:49	<b>ThresholdIntType.</b> Reset: Cold,0h. Specifies the type of interrupt signaled when Ovrlw is set and IntP==1. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[ThresholdLvtOffset]). 10b = SMI trigger event. 11b = Reserved. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::L2::MCA_MISC0_L2[Locked]) ? Read-write : Read-only.
48	<b>Ovrlw.</b> Reset: Cold,0. Set by hardware when ErrCnt transitions from FFEh to FFFh. When this field is set, ErrCnt no longer increments. When this bit is set, the interrupt selected by the ThresholdIntType field is generated. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::L2::MCA_MISC0_L2[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	<b>ErrCnt.</b> Reset: Cold,000h. This is written by software to set the starting value of the error counter. This is incremented by hardware when errors are logged. When this counter overflows, it stays at FFFh (no rollover). The threshold value, written by software, is (FFFh - the desired error count (the number of errors necessary in order for an interrupt to be taken)); the desired error count of 0 (a write value of FFFh) is not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::L2::MCA_MISC0_L2[Locked]) ? Read-write : Read-only.
31:24	<b>BlkPtr.</b> Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

**MSRC000\_2024 [L2 Machine Check Configuration Thread 0] (MCA::L2::MCA\_CONFIG\_L2)**

Reset: 0000\_0000\_0000\_0125h.

Controls configuration of the associated machine check bank.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst2\_n[23:0]\_aliasMSR; MSRC000\_2024

Bits	Description
63:41	Reserved.
40	<b>IntEn.</b> Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	<b>DeferredIntType.</b> Read-write. Reset: 0h. Specifies the type of interrupt signaled when a deferred error is logged. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[DeferredLvtOffset]). 10b = SMI trigger event. 11b = Reserved.
36:35	Reserved.
34	<b>LogDeferredInMcaStat.</b> Read-write. Reset: 0. Init: BIOS,1. 1=Log deferred errors in MCA::L2::MCA_STATUS_L2 and MCA::L2::MCA_ADDR_L2 in addition to MCA::L2::MCA_DESTAT_L2 and MCA::L2::MCA_DEADDR_L2. 0=Only log deferred errors in MCA::L2::MCA_DESTAT_L2 and MCA::L2::MCA_DEADDR_L2. This bit does not affect logging of deferred errors in MCA::L2::MCA_SYND_L2, MCA::L2::MCA_MISC0_L2.
33	Reserved.
32	<b>McaXEnable.</b> Read-write. Reset: 0. Init: BIOS,1. Check: 1. 1=Software has acknowledged support for the MCAX feature set. 0=Software has not acknowledged support for the MCAX feature set. All uncorrected and fatal errors will cause an ErrorEvent packet to be generated. Deferred error interrupts are configured via Core::X86::Msr::McaIntrCfg.
31:11	Reserved.
10	<b>IntPresent.</b> Read-only, Volatile. Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::L2::MCA_CONFIG_L2[IntEn].
9	<b>McaFruTextInMca.</b> Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	<b>McaLsbInStatusSupported.</b> Read-only. Reset: 1. 1=MCA::L2::MCA_CONFIG_L2[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	<b>DeferredIntTypeSupported.</b> Read-only. Reset: 1. 1=MCA::L2::MCA_CONFIG_L2[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::L2::MCA_CONFIG_L2[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	<b>DeferredErrorLoggingSupported.</b> Read-only. Reset: 1. 1=Deferred errors are supported in this MCA bank, and MCA::L2::MCA_CONFIG_L2[LogDeferredInMcaStat] controls the logging behavior of these errors. MCA::L2::MCA_DESTAT_L2 and MCA::L2::MCA_DEADDR_L2 are supported in this MCA bank. 0=Deferred errors are not supported in this bank.
1	Reserved.
0	<b>McaX.</b> Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::L2::MCA_MISC0_L2[BlkPtr] indicates the presence of the additional MISC registers, but is not used to determine their MSR numbers. Deferred error interrupt type is specifiable by MCA bank. MCA::L2::MCA_STATUS_L2[TCC] is present.

**MSRC000\_2025 [L2 IP Identification Thread 0] (MCA::L2::MCA\_IPID\_L2)**

Reset: 0002\_00B0\_0000\_0000h.

The MCA::L2::MCA\_IPID\_L2 register is used by software to determine what IP type and revision is associated with the MCA bank.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst2\_n[23:0]\_aliasMSR; MSRC000\_2025

Bits	Description
63:48	<b>McaType</b> . Read-only. Reset: 0002h. The McaType of the MCA bank within this IP.
47:44	<b>InstanceIdHi</b> . Read-write. Reset: 0h. The HI value instance ID of this IP. This is initialized to a unique ID per instance of this register.
43:32	<b>HardwareID</b> . Read-only. Reset: 0B0h. The Hardware ID of the IP associated with this MCA bank.
31:0	<b>InstanceId</b> . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _lthree0_core0_thread0_inst2_n0_aliasMSR: PSP,2080_8A00h
	Init: _lthree0_core0_thread1_inst2_n1_aliasMSR: PSP,2080_8B00h
	Init: _lthree0_core1_thread0_inst2_n2_aliasMSR: PSP,2082_8A00h
	Init: _lthree0_core1_thread1_inst2_n3_aliasMSR: PSP,2082_8B00h
	Init: _lthree0_core2_thread0_inst2_n4_aliasMSR: PSP,2084_8A00h
	Init: _lthree0_core2_thread1_inst2_n5_aliasMSR: PSP,2084_8B00h
	Init: _lthree0_core3_thread0_inst2_n6_aliasMSR: PSP,2086_8A00h
	Init: _lthree0_core3_thread1_inst2_n7_aliasMSR: PSP,2086_8B00h
	Init: _lthree1_core0_thread0_inst2_n8_aliasMSR: PSP,20C0_8A00h
	Init: _lthree1_core0_thread1_inst2_n9_aliasMSR: PSP,20C0_8B00h
	Init: _lthree1_core1_thread0_inst2_n10_aliasMSR: PSP,20C2_8A00h
	Init: _lthree1_core1_thread1_inst2_n11_aliasMSR: PSP,20C2_8B00h
	Init: _lthree1_core2_thread0_inst2_n12_aliasMSR: PSP,20C4_8A00h
	Init: _lthree1_core2_thread1_inst2_n13_aliasMSR: PSP,20C4_8B00h
	Init: _lthree1_core3_thread0_inst2_n14_aliasMSR: PSP,20C6_8A00h
	Init: _lthree1_core3_thread1_inst2_n15_aliasMSR: PSP,20C6_8B00h
	Init: _lthree1_core4_thread0_inst2_n16_aliasMSR: PSP,20C8_8A00h
	Init: _lthree1_core4_thread1_inst2_n17_aliasMSR: PSP,20C8_8B00h
	Init: _lthree1_core5_thread0_inst2_n18_aliasMSR: PSP,20CA_8A00h
	Init: _lthree1_core5_thread1_inst2_n19_aliasMSR: PSP,20CA_8B00h
	Init: _lthree1_core6_thread0_inst2_n20_aliasMSR: PSP,20CC_8A00h
	Init: _lthree1_core6_thread1_inst2_n21_aliasMSR: PSP,20CC_8B00h
	Init: _lthree1_core7_thread0_inst2_n22_aliasMSR: PSP,20CE_8A00h
	Init: _lthree1_core7_thread1_inst2_n23_aliasMSR: PSP,20CE_8B00h

**MSRC000\_2026 [L2 Machine Check Syndrome Thread 0] (MCA::L2::MCA\_SYND\_L2)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::L2::MCA\_STATUS\_L2 Thread 0

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst2\_n[23:0]\_aliasMSR; MSRC000\_2026

Bits	Description
63:32	<b>Syndrome</b> . Read-write, Volatile. Reset: Cold, 0000_0000h. Contains the syndrome, if any, associated with the error logged in MCA::L2::MCA_STATUS_L2. The low-order bit of the syndrome is stored in bit 0, and the syndrome has a length specified by MCA::L2::MCA_SYND_L2[Length]. The Syndrome field is only valid when MCA::L2::MCA_SYND_L2[Length] is not 0.
31:27	Reserved.
26:24	<b>ErrorPriority</b> . Read-write, Volatile. Reset: Cold, 0h. Encodes the priority of the error logged in MCA::L2::MCA_SYND_L2. 3'b000 = No error; 3'b001 = Reserved; 3'b010 = Corrected Error; 3'b011 = Deferred Error; 3'b100 = Uncorrected Error; 3'b101 = Fatal Error; all others reserved.
23:18	<b>Length</b> . Read-write, Volatile. Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::L2::MCA_SYND_L2[Syndrome]. A value of 0 indicates that there is no valid syndrome in MCA::L2::MCA_SYND_L2. For example, a syndrome length of 9 means that MCA::L2::MCA_SYND_L2[Syndrome] bits [8:0] contains a valid syndrome.
17:0	<b>ErrorInformation</b> . Read-write, Volatile. Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 45 [MCA_SYND_L2].

Table 45: MCA\_SYND\_L2

Error Type	Bits	Description
MultiHit	[18:8] [7:0]	Index One-hot way vector
Tag	[17:14] [13:3] [2:0]	Reserved Index Way
Data	[17:17] [16] [15:5] [4] [3] [2:0]	Reserved Poison Index TopBotLoc Reserved Way
Hwa	[17:0]	Reserved
Sdp	[17:17] [16:6] [5:3] [2:0]	Reserved Addr[16:6] Addr[5:3] L2Way
StateMachine	[17:2] [1] [0]	Reserved Reserved Reserved
Wdt	[17:0]	Reserved

**MSRC000\_2028 [L2 Machine Check Deferred Error Status Thread 0] (MCA::L2::MCA\_DESTAT\_L2)**

Reset: Cold,0000\_0000\_0000\_0000h.

Holds status information for the first deferred error seen in this bank.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst2\_n[23:0]\_aliasMSR; MSRC000\_2028

Bits	Description
63	<b>Val.</b> Read-write, Volatile. Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	<b>Overflow.</b> Read-write, Volatile. Reset: Cold,0. 1=An error was detected while the valid bit (Val) was set; at least one error was not logged. Overflow is set independently of whether the existing error is overwritten. (See the section on error overwrite priorities.)
61:59	<b>RESERV4.</b> Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	<b>AddrV.</b> Read-write, Volatile. Reset: Cold,0. 1=MCA::L2::MCA_DEADDR_L2 contains address information associated with the error.
57:54	<b>RESERV3.</b> Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	<b>SyndV.</b> Read-write, Volatile. Reset: Cold,0. 1=This error logged information in MCA::L2::MCA_SYND_L2. If MCA::L2::MCA_SYND_L2[ErrorPriority] is the same as the priority of the error in MCA::L2::MCA_STATUS_L2, then the information in MCA::L2::MCA_SYND_L2 is associated with the error in MCA::L2::MCA_DESTAT_L2.
52:45	<b>RESERV2.</b> Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	<b>Deferred.</b> Read-write, Volatile. Reset: Cold,0. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; poison is created and an exception is deferred until the poison data is consumed.
43:30	<b>RESERV1.</b> Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	<b>AddrLsb.</b> Read-write, Volatile. Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::L2::MCA_ADDR_L2[ErrorAddr]. A value of 0 indicates that MCA::L2::MCA_ADDR_L2[63:0] contains a valid byte address. A value of 6 indicates that MCA::L2::MCA_ADDR_L2[63:6] contains a valid cache line address and that MCA::L2::MCA_ADDR_L2[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::L2::MCA_ADDR_L2[63:12] contain a valid 4KB memory page and that MCA::L2::MCA_ADDR_L2[11:0] should be ignored by error handling software.
23:22	<b>RESERV0.</b> Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	<b>ErrorCodeExt.</b> Read-write, Volatile. Reset: Cold,00h. Logs an extended error code when an error is detected. This model-specific field is used in conjunction with ErrorCode to identify the error sub-type for root cause analysis.
15:0	<b>ErrorCode.</b> Read-write, Volatile. Reset: Cold,0000h. Error code for this error.

**MSRC000\_2029 [L2 Deferred Error Address Thread 0] (MCA::L2::MCA\_DEADDR\_L2)**

Read-write, Volatile. Reset: Cold,0000\_0000\_0000\_0000h.

The MCA::L2::MCA\_DEADDR\_L2 register stores the address associated with the error in MCA::L2::MCA\_DESTAT\_L2. The register is only meaningful if MCA::L2::MCA\_DESTAT\_L2[Val]=1 and MCA::L2::MCA\_DESTAT\_L2[AddrV]=1. The lowest valid bit of the address is defined by MCA::L2::MCA\_DESTAT\_L2[AddrLsb].

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst2\_n[23:0]\_aliasMSR; MSRC000\_2029

Bits	Description
63:0	<b>ErrorAddr.</b> Read-write, Volatile. Reset: Cold,0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::L2::MCA_DESTAT_L2. The lowest-order valid bit of the address is specified in MCA::L2::MCA_DESTAT_L2[AddrLsb].

**MSRC001\_0402 [L2 Machine Check Control Mask Thread 0] (MCA::L2::MCA\_CTL\_MASK\_L2)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

Inhibit detection of an error source.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst2\_n[23:0]\_aliasMSR; MSRC001\_0402

Bits	Description
63:7	Reserved.
6	<b>Wdt.</b> Read-write. Reset: 0. Reserved
5	<b>StateMachine.</b> Read-write. Reset: 0. Error initiated by programmable state machine.
4	<b>Sdp.</b> Read-write. Reset: 0. SDP Read Response Parity Error
3	<b>Hwa.</b> Read-write. Reset: 0. Hardware Assert Error.
2	<b>Data.</b> Read-write. Reset: 0. L2M Data Array ECC Error.
1	<b>Tag.</b> Read-write. Reset: 0. L2M Tag or State Array ECC Error.
0	<b>MultiHit.</b> Read-write. Reset: 0. L2M Tag Multiple-Way-Hit error.

**MSRC000\_202E [L2 Machine Check Syndrome Extended Thread 0] (MCA::L2::MCA\_SYND1\_L2)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::L2::MCA\_STATUS\_L2 Thread 0

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst2\_n[23:0]\_aliasMSR; MSRC000\_202E

Bits	Description
63:0	<b>Syndrome.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::L2::MCA_SYND1_L2 register stores information associated with the error in MCA::L2::MCA_STATUS_L2 or MCA_DESTAT. The register is meaningful if MCA::L2::MCA_STATUS_L2[SyndV]=1. When MCA::L2::MCA_CONFIG_L2[McaFruTextInMca]=1, MCA::L2::MCA_SYND1_L2 stores ASCII FruText associated with the error.

**MSRC000\_202F [L2 Machine Check Syndrome Extended Thread 0] (MCA::L2::MCA\_SYND2\_L2)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::L2::MCA\_STATUS\_L2 Thread 0

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst2\_n[23:0]\_aliasMSR; MSRC000\_202F

Bits	Description
63:0	<b>Syndrome.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::L2::MCA_SYND2_L2 register stores information associated with the error in MCA::L2::MCA_STATUS_L2 or MCA_DESTAT. The register is meaningful if MCA::L2::MCA_STATUS_L2[SyndV]=1. When MCA::L2::MCA_CONFIG_L2[McaFruTextInMca]=1, MCA::L2::MCA_SYND2_L2 stores ASCII FruText associated with the error.

## 3.2.5.4 DE

**MSR0000\_040C...MSRC000\_2030 [DE Machine Check Control Thread 0] (MCA::DE::MCA\_CTL\_DE)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

0=Disables error reporting for the corresponding error. 1=Enables error reporting via machine check exception for the corresponding error. The MCA::DE::MCA\_CTL\_DE register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG\_CTL. Does not affect error detection, correction, or logging.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst3\_n[23:0]\_aliasMSRLEGACY; MSR0000\_040C

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst3\_n[23:0]\_aliasMSR; MSRC000\_2030

Bits	Description
63:10	Reserved.
9	<b>HWA.</b> Read-write. Reset: 0. Hardware Assertion Error.
8	<b>OCQ.</b> Read-write. Reset: 0. Micro-op fetch queue parity error.
7	<b>UcSeq.</b> Read-write. Reset: 0. Patch RAM sequencer parity error.
6	<b>UcDat.</b> Read-write. Reset: 0. Patch RAM data parity error.
5	<b>Faq.</b> Read-write. Reset: 0. Fetch address FIFO parity error.
4	<b>Idq.</b> Read-write. Reset: 0. Instruction dispatch queue parity error.
3	<b>UopQ.</b> Read-write. Reset: 0. Micro-op queue parity error.
2	<b>Ibq.</b> Read-write. Reset: 0. Instruction buffer parity error.
1	<b>OcDat.</b> Read-write. Reset: 0. Micro-op cache data parity error.
0	<b>OcTag.</b> Read-write. Reset: 0. Micro-op cache tag parity error.



**MSR0000\_040D...MSRC000\_2031 [DE Machine Check Status Thread 0] (MCA::DE::MCA\_STATUS\_DE)**

Reset: Cold,0000\_0000\_0000\_0000h.

Logs information associated with errors.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst3\_n[23:0]\_aliasMSRLEGACY; MSR0000\_040D

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst3\_n[23:0]\_aliasMSR; MSRC000\_2031

Bits	Description
63	<b>Val.</b> Reset: Cold,0. 1=A valid error has been detected. This bit should be cleared by software after the register has been read. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
62	<b>Overflow.</b> Reset: Cold,0. 1=An error was detected while the valid bit (Val) was set; at least one error was not logged. Overflow is set independently of whether the existing error is overwritten. See 3.1.3 [Machine Check Errors]. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
61	<b>UC.</b> Reset: Cold,0. 1=The error was not corrected by hardware. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
60	<b>En.</b> Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::DE::MCA_CTL_DE. This bit is a copy of bit in MCA::DE::MCA_CTL_DE for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	<b>MiscV.</b> Reset: Cold,0. 1=Valid thresholding in MCA::DE::MCA_MISC0_DE. In certain modes, MISC registers are owned by platform firmware and will RAZ when read by non-SMM code. Therefore, it is possible for MiscV=1 and the MISC register to read as all zeros. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
58	<b>AddrV.</b> Reset: Cold,0. 1=MCA::DE::MCA_ADDR_DE contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	<b>PCC.</b> Reset: Cold,0. 1=Hardware context held by the processor may have been corrupted. Continued operation of the system may have unpredictable results. The error is not recoverable or survivable, and the system should be reinitialized. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
56	<b>ErrCoreIdVal.</b> Reset: Cold,0. 1=The ErrCoreId field is valid. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
55	<b>TCC.</b> Reset: Cold,0. 1=Hardware context of the process thread to which the error was reported may have been corrupted. Continued operation of the thread may have unpredictable results. The thread must be terminated. Only meaningful when MCA::DE::MCA_STATUS_DE[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	<b>RESERV54.</b> Reset: Cold,0. MCA_STATUS Register Reserved bit. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
53	<b>SyndV.</b> Reset: Cold,0. 1=This error logged information in MCA::DE::MCA_SYND_DE. If MCA::DE::MCA_SYND_DE[ErrorPriority] is the same as the priority of the error in MCA::DE::MCA_STATUS_DE, then the information in MCA::DE::MCA_SYND_DE is associated with the error in MCA::DE::MCA_STATUS_DE. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	<b>RESERV47.</b> Reset: Cold,00h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
46	<b>CECC.</b> Reset: Cold,0. 1=The error was a correctable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
45	<b>UECC.</b> Reset: Cold,0. 1=The error was an uncorrectable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.



44	<b>Deferred.</b> Reset: Cold,0. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; an exception is deferred until the erroneous data is consumed. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
43	<b>Poison.</b> Reset: Cold,0. 1=The error was the result of attempting to consume poisoned data. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
42:41	<b>RESERV41.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
40	<b>Scrub.</b> Reset: Cold,0. 1=The error was the result of a scrub operation. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
39:38	<b>RESERV38.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
37:32	<b>ErrCoreId.</b> Reset: Cold,00h. When ErrCoreIdVal=1 this field indicates which core within the processor is associated with the error; Otherwise this field is reserved. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
31:30	<b>RESERV30.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
29:24	<b>AddrLsb.</b> Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::DE::MCA_ADDR_DE[ErrorAddr]. A value of 0 indicates that MCA::DE::MCA_ADDR_DE[63:0] contains a valid byte address. A value of 6 indicates that MCA::DE::MCA_ADDR_DE[63:6] contains a valid cache line address and that MCA::DE::MCA_ADDR_DE[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::DE::MCA_ADDR_DE[63:12] contain a valid 4KB memory page and that MCA::DE::MCA_ADDR_DE[11:0] should be ignored by error handling software. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
23:22	<b>RESERV22.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
21:16	<b>ErrorCodeExt.</b> Reset: Cold,00h. Extended Error Code. This field is used to identify the error type for root cause analysis. This field indicates which bit position in MCA::DE::MCA_CTL_DE enables error reporting for the logged error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
15:0	<b>ErrorCode.</b> Reset: Cold,0000h. Error code for this error. See 3.1.3.3 [Error Codes] for details on decoding this field. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.

Table 46: MCA\_STATUS\_DE

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
OcTag	0x0	0	0	0	0	0	0
OcDat	0x1	0	0	0	0	0	0
Ibq	0x2	1	1	1	0	0	0
UopQ	0x3	1	1	1	0	0	0
Idq	0x4	1	1	1	0	0	0
Faq	0x5	1	1	1	0	0	0
UcDat	0x6	1	1	1	0	0	0
UcSeq	0x7	1	1	1	0	0	0
OCQ	0x8	0	0	0	0	0	0
HWA	0x9	1	1	1	0	0	0
Reserved	0xa	0	0	0	0	0	1

**MSR0000\_040E...MSRC000\_2032 [DE Machine Check Address Thread 0] (MCA::DE::MCA\_ADDR\_DE)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

MCA::DE::MCA\_ADDR\_DE stores an address and other information associated with the error in MCA::DE::MCA\_STATUS\_DE. The register is only meaningful if MCA::DE::MCA\_STATUS\_DE[Val]=1 and MCA::DE::MCA\_STATUS\_DE[AddrV]=1.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst3\_n[23:0]\_aliasMSRLEGACY; MSR0000\_040E

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst3\_n[23:0]\_aliasMSR; MSRC000\_2032

Bits	Description
63:0	<b>ErrorAddr.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. Unless otherwise specified by an error, contains the address associated with the error logged in MCA::DE::MCA_STATUS_DE. For physical addresses, the most significant bit is given by Core::X86::Cpuid::LongModeInfo[PhysAddrSize].

Table 47: MCA\_ADDR\_DE

Error Type	Bits	Description
OcTag	[63:0]	Reserved
OcDat	[63:0]	Reserved
Ibq	[63:0]	Reserved
UopQ	[63:0]	Reserved
Idq	[63:0]	Reserved
Faq	[63:0]	Reserved
UcDat	[63:0]	Reserved
UcSeq	[63:0]	Reserved
OCQ	[63:0]	Reserved
HWA	[63:0]	Reserved
Reserved	[63:0]	Reserved

**MSR0000\_040F...MSRC000\_2033 [DE Machine Check Miscellaneous 0 Thread 0]  
(MCA::DE::MCA\_MISC0\_DE)**

Log miscellaneous information associated with errors.	
_lthree[1:0]_core[7:0]_thread[1:0]_inst3_n[23:0]_aliasMSRLEGACY; MSR0000_040F	
_lthree[1:0]_core[7:0]_thread[1:0]_inst3_n[23:0]_aliasMSR; MSRC000_2033	
Bits	Description
63	<b>Valid.</b> Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	<b>CntP.</b> Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	<b>Locked.</b> Reset: 0. 1=Writes to this register are ignored. This bit is set by BIOS to indicate that this register is not available for OS use. BIOS should set this bit if ThresholdIntType is set to SMI. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read-only.
60	<b>IntP.</b> Reset: 1. 1=ThresholdIntType can be used to generate interrupts. 0=ThresholdIntType and interrupt generation are not supported. AccessType: (Core::X86::Msrr::HWCR[McStatusWrEn]   !MCA::DE::MCA_MISC0_DE[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	<b>LvtOffset.</b> Reset: 0h. One per die. For error thresholding interrupts, specifies the address of the LVT entry in the APIC registers as follows: LVT address = (LvtOffset shifted left 4 bits) + 500h (see Core::X86::Apic::ExtendedInterruptLvtEntries). AccessType: (Core::X86::Msrr::HWCR[McStatusWrEn]   !MCA::DE::MCA_MISC0_DE[Locked]) ? Read-write : Read-only.
51	<b>CntEn.</b> Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msrr::HWCR[McStatusWrEn]   !MCA::DE::MCA_MISC0_DE[Locked]) ? Read-write : Read-only.
50:49	<b>ThresholdIntType.</b> Reset: Cold,0h. Specifies the type of interrupt signaled when Ovrflw is set and IntP==1. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msrr::McaIntrCfg[ThresholdLvtOffset]). 10b = SMI trigger event. 11b = Reserved. AccessType: (Core::X86::Msrr::HWCR[McStatusWrEn]   !MCA::DE::MCA_MISC0_DE[Locked]) ? Read-write : Read-only.
48	<b>Ovrflw.</b> Reset: Cold,0. Set by hardware when ErrCnt transitions from FFEh to FFFh. When this field is set, ErrCnt no longer increments. When this bit is set, the interrupt selected by the ThresholdIntType field is generated. AccessType: (Core::X86::Msrr::HWCR[McStatusWrEn]   !MCA::DE::MCA_MISC0_DE[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	<b>ErrCnt.</b> Reset: Cold,000h. This is written by software to set the starting value of the error counter. This is incremented by hardware when errors are logged. When this counter overflows, it stays at FFFh (no rollover). The threshold value, written by software, is (FFFh - the desired error count (the number of errors necessary in order for an interrupt to be taken)); the desired error count of 0 (a write value of FFFh) is not supported. AccessType: (Core::X86::Msrr::HWCR[McStatusWrEn]   !MCA::DE::MCA_MISC0_DE[Locked]) ? Read-write : Read-only.
31:24	<b>BlkPtr.</b> Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

**MSRC000\_2034 [DE Machine Check Configuration Thread 0] (MCA::DE::MCA\_CONFIG\_DE)**

Reset: 0000\_0000\_0000\_0121h.

Controls configuration of the associated machine check bank.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst3\_n[23:0]\_aliasMSR; MSRC000\_2034

Bits	Description
63:41	Reserved.
40	<b>IntEn.</b> Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	<b>DeferredIntType.</b> Read-write. Reset: 0h. Specifies the type of interrupt signaled when a deferred error is logged. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[DeferredLvtOffset]). 10b = SMI trigger event. 11b = Reserved.
36:33	Reserved.
32	<b>McaXEnable.</b> Read-write. Reset: 0. Init: BIOS,1. Check: 1. 1=Software has acknowledged support for the MCAX feature set. 0=Software has not acknowledged support for the MCAX feature set. All uncorrected and fatal errors will cause an ErrorEvent packet to be generated. Deferred error interrupts are configured via Core::X86::Msr::McaIntrCfg.
31:11	Reserved.
10	<b>IntPresent.</b> Read-only, Volatile. Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::DE::MCA_CONFIG_DE[IntEn].
9	<b>McaFruTextInMca.</b> Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	<b>McaLsbInStatusSupported.</b> Read-only. Reset: 1. 1=MCA::DE::MCA_CONFIG_DE[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	<b>DeferredIntTypeSupported.</b> Read-only. Reset: 1. 1=MCA::DE::MCA_CONFIG_DE[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::DE::MCA_CONFIG_DE[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	<b>DeferredErrorLoggingSupported.</b> Read-only. Reset: 0. 1=Deferred errors are supported in this MCA bank, and the LogDeferredInMcaStat field in this register controls the logging behavior of these errors. 0=Deferred errors are not supported in this bank.
1	Reserved.
0	<b>McaX.</b> Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::DE::MCA_MISC0_DE[BlkPtr] indicates the presence of the additional MISC registers, but is not used to determine their MSR numbers. Deferred error interrupt type is specifiable by MCA bank. MCA::DE::MCA_STATUS_DE[TCC] is present.

**MSRC000\_2035 [DE IP Identification Thread 0] (MCA::DE::MCA\_IPID\_DE)**

Reset: 0003\_00B0\_0000\_0000h.

The MCA::DE::MCA\_IPID\_DE register is used by software to determine what IP type and revision is associated with the MCA bank.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst3\_n[23:0]\_aliasMSR; MSRC000\_2035

Bits	Description
63:48	<b>McaType</b> . Read-only. Reset: 0003h. The McaType of the MCA bank within this IP.
47:44	<b>InstanceIdHi</b> . Read-write. Reset: 0h. The HI value instance ID of this IP. This is initialized to a unique ID per instance of this register.
43:32	<b>HardwareID</b> . Read-only. Reset: 0B0h. The Hardware ID of the IP associated with this MCA bank.
31:0	<b>InstanceId</b> . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _lthree0_core0_thread0_inst3_n0_aliasMSR: PSP,2080_9600h
	Init: _lthree0_core0_thread1_inst3_n1_aliasMSR: PSP,2080_9700h
	Init: _lthree0_core1_thread0_inst3_n2_aliasMSR: PSP,2082_9600h
	Init: _lthree0_core1_thread1_inst3_n3_aliasMSR: PSP,2082_9700h
	Init: _lthree0_core2_thread0_inst3_n4_aliasMSR: PSP,2084_9600h
	Init: _lthree0_core2_thread1_inst3_n5_aliasMSR: PSP,2084_9700h
	Init: _lthree0_core3_thread0_inst3_n6_aliasMSR: PSP,2086_9600h
	Init: _lthree0_core3_thread1_inst3_n7_aliasMSR: PSP,2086_9700h
	Init: _lthree1_core0_thread0_inst3_n8_aliasMSR: PSP,20C0_9600h
	Init: _lthree1_core0_thread1_inst3_n9_aliasMSR: PSP,20C0_9700h
	Init: _lthree1_core1_thread0_inst3_n10_aliasMSR: PSP,20C2_9600h
	Init: _lthree1_core1_thread1_inst3_n11_aliasMSR: PSP,20C2_9700h
	Init: _lthree1_core2_thread0_inst3_n12_aliasMSR: PSP,20C4_9600h
	Init: _lthree1_core2_thread1_inst3_n13_aliasMSR: PSP,20C4_9700h
	Init: _lthree1_core3_thread0_inst3_n14_aliasMSR: PSP,20C6_9600h
	Init: _lthree1_core3_thread1_inst3_n15_aliasMSR: PSP,20C6_9700h
	Init: _lthree1_core4_thread0_inst3_n16_aliasMSR: PSP,20C8_9600h
	Init: _lthree1_core4_thread1_inst3_n17_aliasMSR: PSP,20C8_9700h
	Init: _lthree1_core5_thread0_inst3_n18_aliasMSR: PSP,20CA_9600h
	Init: _lthree1_core5_thread1_inst3_n19_aliasMSR: PSP,20CA_9700h
	Init: _lthree1_core6_thread0_inst3_n20_aliasMSR: PSP,20CC_9600h
	Init: _lthree1_core6_thread1_inst3_n21_aliasMSR: PSP,20CC_9700h
	Init: _lthree1_core7_thread0_inst3_n22_aliasMSR: PSP,20CE_9600h
	Init: _lthree1_core7_thread1_inst3_n23_aliasMSR: PSP,20CE_9700h

**MSRC000\_2036 [DE Machine Check Syndrome Thread 0] (MCA::DE::MCA\_SYND\_DE)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::DE::MCA\_STATUS\_DE Thread 0

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst3\_n[23:0]\_aliasMSR; MSRC000\_2036

Bits	Description
63:32	<b>Syndrom</b> . Read-write, Volatile. Reset: Cold, 0000_0000h. Contains the syndrome, if any, associated with the error logged in MCA::DE::MCA_STATUS_DE. The low-order bit of the syndrome is stored in bit 0, and the syndrome has a length specified by MCA::DE::MCA_SYND_DE[Length]. The Syndrome field is only valid when MCA::DE::MCA_SYND_DE[Length] is not 0.
31:27	Reserved.
26:24	<b>ErrorPriority</b> . Read-write, Volatile. Reset: Cold, 0h. Encodes the priority of the error logged in MCA::DE::MCA_SYND_DE. 3'b000 = No error; 3'b001 = Reserved; 3'b010 = Corrected Error; 3'b011 = Deferred Error; 3'b100 = Uncorrected Error; 3'b101 = Fatal Error; all others reserved.
23:18	<b>Length</b> . Read-write, Volatile. Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::DE::MCA_SYND_DE[Syndrome]. A value of 0 indicates that there is no valid syndrome in MCA::DE::MCA_SYND_DE. For example, a syndrome length of 9 means that MCA::DE::MCA_SYND_DE[Syndrome] bits [8:0] contains a valid syndrome.
17:0	<b>ErrorInformation</b> . Read-write, Volatile. Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 48 [MCA_SYND_DE].

Table 48: MCA\_SYND\_DE

Error Type	Bits	Description
OcTag	[17:10] [9:6] [5:0]	Reserved Way Index
OcDat	[17:10] [9:6] [5:0]	Reserved Way Index
Ibq	[17:0]	Reserved
UopQ	[17:0]	Reserved
Idq	[17:0]	Reserved
Faq	[17:0]	Reserved
UcDat	[17:0]	Reserved
UcSeq	[17:0]	Reserved
OCQ	[17:0]	Reserved
HWA	[17:6] [5:0]	Reserved Reserved
Reserved	[17:0]	Reserved

**MSRC001\_0403 [DE Machine Check Control Mask Thread 0] (MCA::DE::MCA\_CTL\_MASK\_DE)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

Inhibit detection of an error source.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst3\_n[23:0]\_aliasMSR; MSRC001\_0403

Bits	Description
63:10	Reserved.
9	<b>HWA.</b> Read-write. Reset: 0. Hardware Assertion Error.
8	<b>OCQ.</b> Read-write. Reset: 0. Micro-op fetch queue parity error.
7	<b>UcSeq.</b> Read-write. Reset: 0. Patch RAM sequencer parity error.
6	<b>UcDat.</b> Read-write. Reset: 0. Patch RAM data parity error.
5	<b>Faq.</b> Read-write. Reset: 0. Fetch address FIFO parity error.
4	<b>Idq.</b> Read-write. Reset: 0. Instruction dispatch queue parity error.
3	<b>UopQ.</b> Read-write. Reset: 0. Micro-op queue parity error.
2	<b>Ibq.</b> Read-write. Reset: 0. Instruction buffer parity error.
1	<b>OcDat.</b> Read-write. Reset: 0. Micro-op cache data parity error.
0	<b>OcTag.</b> Read-write. Reset: 0. Micro-op cache tag parity error.

**MSRC000\_203E [DE Machine Check Syndrome Extended Thread 0] (MCA::DE::MCA\_SYND1\_DE)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::DE::MCA\_STATUS\_DE Thread 0

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst3\_n[23:0]\_aliasMSR; MSRC000\_203E

Bits	Description
63:0	<b>Syndrome.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::DE::MCA_SYND1_DE register stores information associated with the error in MCA::DE::MCA_STATUS_DE or MCA_DESTAT. The register is meaningful if MCA::DE::MCA_STATUS_DE[SyndV]=1. When MCA::DE::MCA_CONFIG_DE[McaFruTextInMca]=1, MCA::DE::MCA_SYND1_DE stores ASCII FruText associated with the error.

**MSRC000\_203F [DE Machine Check Syndrome Extended Thread 0] (MCA::DE::MCA\_SYND2\_DE)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::DE::MCA\_STATUS\_DE Thread 0

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst3\_n[23:0]\_aliasMSR; MSRC000\_203F

Bits	Description
63:0	<b>Syndrome.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::DE::MCA_SYND2_DE register stores information associated with the error in MCA::DE::MCA_STATUS_DE or MCA_DESTAT. The register is meaningful if MCA::DE::MCA_STATUS_DE[SyndV]=1. When MCA::DE::MCA_CONFIG_DE[McaFruTextInMca]=1, MCA::DE::MCA_SYND2_DE stores ASCII FruText associated with the error.

**3.2.5.5 EX****MSR0000\_0414...MSRC000\_2050 [EX Machine Check Control Thread 0] (MCA::EX::MCA\_CTL\_EX)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

0=Disables error reporting for the corresponding error. 1=Enables error reporting via machine check exception for the corresponding error. The MCA::EX::MCA\_CTL\_EX register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG\_CTL. Does not affect error detection, correction, or logging.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst5\_n[23:0]\_aliasMSRLEGACY; MSR0000\_0414

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst5\_n[23:0]\_aliasMSR; MSRC000\_2050

Bits	Description
63:14	Reserved.
13	<b>RETMAP.</b> Read-write. Reset: 0. Retire Map parity error.
12	<b>SPECMAP.</b> Read-write. Reset: 0. PRN/FRN freelist parity error.
11	<b>HWA.</b> Read-write. Reset: 0. Hardware Assertion error.
10	<b>BBQ.</b> Read-write. Reset: 0. Branch buffer queue parity error.
9	<b>SQ.</b> Read-write. Reset: 0. EXTID parity error.
8	<b>STATQ.</b> Read-write. Reset: 0. Retire status queue parity error.
7	<b>RETDISP.</b> Read-write. Reset: 0. Retire dispatch queue parity error.
6	<b>CHKPTQ.</b> Read-write. Reset: 0. CHKPTQ. Checkpoint queue parity error.
5	<b>PLDAL.</b> Read-write. Reset: 0. EX payload parity error.
4	<b>PLDAG.</b> Read-write. Reset: 0. Address generator payload parity error.
3	<b>IDRF.</b> Read-write. Reset: 0. Immediate displacement register file parity error.
2	<b>FRF.</b> Read-write. Reset: 0. Flag register file parity error.
1	<b>PRF.</b> Read-write. Reset: 0. Physical register file parity error.
0	<b>WDT.</b> Read-write. Reset: 0. Watchdog Timeout error.



**MSR0000\_0415...MSRC000\_2051 [EX Machine Check Status Thread 0] (MCA::EX::MCA\_STATUS\_EX)**

Reset: Cold,0000\_0000\_0000\_0000h.

Logs information associated with errors.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst5\_n[23:0]\_aliasMSRLEGACY; MSR0000\_0415

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst5\_n[23:0]\_aliasMSR; MSRC000\_2051

Bits	Description
63	<b>Val.</b> Reset: Cold,0. 1=A valid error has been detected. This bit should be cleared by software after the register has been read. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
62	<b>Overflow.</b> Reset: Cold,0. 1=An error was detected while the valid bit (Val) was set; at least one error was not logged. Overflow is set independently of whether the existing error is overwritten. See 3.1.3 [Machine Check Errors]. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
61	<b>UC.</b> Reset: Cold,0. 1=The error was not corrected by hardware. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
60	<b>En.</b> Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::EX::MCA_CTL_EX. This bit is a copy of bit in MCA::EX::MCA_CTL_EX for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	<b>MiscV.</b> Reset: Cold,0. 1=Valid thresholding in MCA::EX::MCA_MISC0_EX. In certain modes, MISC registers are owned by platform firmware and will RAZ when read by non-SMM code. Therefore, it is possible for MiscV=1 and the MISC register to read as all zeros. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
58	<b>AddrV.</b> Reset: Cold,0. 1=MCA::EX::MCA_ADDR_EX contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	<b>PCC.</b> Reset: Cold,0. 1=Hardware context held by the processor may have been corrupted. Continued operation of the system may have unpredictable results. The error is not recoverable or survivable, and the system should be reinitialized. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
56	<b>ErrCoreIdVal.</b> Reset: Cold,0. 1=The ErrCoreId field is valid. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
55	<b>TCC.</b> Reset: Cold,0. 1=Hardware context of the process thread to which the error was reported may have been corrupted. Continued operation of the thread may have unpredictable results. The thread must be terminated. Only meaningful when MCA::EX::MCA_STATUS_EX[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	<b>RESERV54.</b> Reset: Cold,0. MCA_STATUS Register Reserved bit. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
53	<b>SyndV.</b> Reset: Cold,0. 1=This error logged information in MCA::EX::MCA_SYND_EX. If MCA::EX::MCA_SYND_EX[ErrorPriority] is the same as the priority of the error in MCA::EX::MCA_STATUS_EX, then the information in MCA::EX::MCA_SYND_EX is associated with the error in MCA::EX::MCA_STATUS_EX. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	<b>RESERV47.</b> Reset: Cold,00h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
46	<b>CECC.</b> Reset: Cold,0. 1=The error was a correctable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
45	<b>UECC.</b> Reset: Cold,0. 1=The error was an uncorrectable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.



44	<b>Deferred.</b> Reset: Cold,0. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; an exception is deferred until the erroneous data is consumed. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
43	<b>Poison.</b> Reset: Cold,0. 1=The error was the result of attempting to consume poisoned data. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
42:41	<b>RESERV41.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
40	<b>Scrub.</b> Reset: Cold,0. 1=The error was the result of a scrub operation. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
39:38	<b>RESERV38.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
37:32	<b>ErrCoreId.</b> Reset: Cold,00h. When ErrCoreIdVal=1 this field indicates which core within the processor is associated with the error; Otherwise this field is reserved. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
31:30	<b>RESERV30.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
29:24	<b>AddrLsb.</b> Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::EX::MCA_ADDR_EX[ErrorAddr]. A value of 0 indicates that MCA::EX::MCA_ADDR_EX[63:0] contains a valid byte address. A value of 6 indicates that MCA::EX::MCA_ADDR_EX[63:6] contains a valid cache line address and that MCA::EX::MCA_ADDR_EX[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::EX::MCA_ADDR_EX[63:12] contain a valid 4KB memory page and that MCA::EX::MCA_ADDR_EX[11:0] should be ignored by error handling software. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
23:22	<b>RESERV22.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
21:16	<b>ErrorCodeExt.</b> Reset: Cold,00h. Extended Error Code. This field is used to identify the error type for root cause analysis. This field indicates which bit position in MCA::EX::MCA_CTL_EX enables error reporting for the logged error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
15:0	<b>ErrorCode.</b> Reset: Cold,0000h. Error code for this error. See 3.1.3.3 [Error Codes] for details on decoding this field. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.

Table 49: MCA\_STATUS\_EX

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
WDT	0x0	1	1	1	0	0	1
PRF	0x1	1	1	1	0	0	1
FRF	0x2	1	1	1	0	0	1
IDRF	0x3	1	1	1	0	0	1
PLDAG	0x4	1	1	1	0	0	0
PLDAL	0x5	1	1	1	0	0	1
CHKPTQ	0x6	1	1	1	0	0	1
RETDISP	0x7	1	1	1	0	0	1
STATQ	0x8	1	1	1	0	0	0
SQ	0x9	1	1	1	0	0	1
BBQ	0xa	1	1	1	0	0	1
HWA	0xb	1	1	1	0	0	1

SPECMAP	0xc	1	1	1	0	0	1
RETMAP	0xd	1	1	1	0	0	0
Reserved	0xe	0	0	0	0	0	1

**MSR0000\_0416...MSRC000\_2052 [EX Machine Check Address Thread 0] (MCA::EX::MCA\_ADDR\_EX)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

MCA::EX::MCA\_ADDR\_EX stores an address and other information associated with the error in MCA::EX::MCA\_STATUS\_EX. The register is only meaningful if MCA::EX::MCA\_STATUS\_EX[Val]=1 and MCA::EX::MCA\_STATUS\_EX[AddrV]=1.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst5\_n[23:0]\_aliasMSRLEGACY; MSR0000\_0416

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst5\_n[23:0]\_aliasMSR; MSRC000\_2052

Bits	Description
63:0	<b>ErrorAddr.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. Unless otherwise specified by an error, contains the address associated with the error logged in MCA::EX::MCA_STATUS_EX. For physical addresses, the most significant bit is given by Core::X86::CpuId::LongModeInfo[PhysAddrSize].

Table 50: MCA\_ADDR\_EX

Error Type	Bits	Description
WDT	[56:0]	RIP of thread triggering the watchdog timeout
PRF	[63:0]	Reserved
FRF	[63:0]	Reserved
IDRF	[63:0]	Reserved
PLDAG	[63:0]	Reserved
PLDAL	[63:0]	Reserved
CHKPTQ	[63:0]	Reserved
RETDISP	[63:0]	Reserved
STATQ	[63:0]	Reserved
SQ	[63:0]	Reserved
BBQ	[63:0]	Reserved
HWA	[63:0]	Reserved
SPECMAP	[63:0]	Reserved
RETMAP	[56:0]	Reserved
Reserved	[63:0]	Reserved

**MSR0000\_0417...MSRC000\_2053 [EX Machine Check Miscellaneous 0 Thread 0]  
(MCA::EX::MCA\_MISC0\_EX)**

Log miscellaneous information associated with errors.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst5\_n[23:0]\_aliasMSRLEGACY; MSR0000\_0417

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst5\_n[23:0]\_aliasMSR; MSRC000\_2053

Bits	Description
63	<b>Valid.</b> Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	<b>CntP.</b> Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	<b>Locked.</b> Reset: 0. 1=Writes to this register are ignored. This bit is set by BIOS to indicate that this register is not available for OS use. BIOS should set this bit if ThresholdIntType is set to SMI. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
60	<b>IntP.</b> Reset: 1. 1=ThresholdIntType can be used to generate interrupts. 0=ThresholdIntType and interrupt generation are not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::EX::MCA_MISC0_EX[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	<b>LvtOffset.</b> Reset: 0h. One per die. For error thresholding interrupts, specifies the address of the LVT entry in the APIC registers as follows: LVT address = (LvtOffset shifted left 4 bits) + 500h (see Core::X86::Apic::ExtendedInterruptLvtEntries). AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::EX::MCA_MISC0_EX[Locked]) ? Read-write : Read-only.
51	<b>CntEn.</b> Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::EX::MCA_MISC0_EX[Locked]) ? Read-write : Read-only.
50:49	<b>ThresholdIntType.</b> Reset: Cold,0h. Specifies the type of interrupt signaled when Ovrflw is set and IntP==1. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[ThresholdLvtOffset]). 10b = SMI trigger event. 11b = Reserved. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::EX::MCA_MISC0_EX[Locked]) ? Read-write : Read-only.
48	<b>Ovrflw.</b> Reset: Cold,0. Set by hardware when ErrCnt transitions from FFEh to FFFh. When this field is set, ErrCnt no longer increments. When this bit is set, the interrupt selected by the ThresholdIntType field is generated. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::EX::MCA_MISC0_EX[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	<b>ErrCnt.</b> Reset: Cold,000h. This is written by software to set the starting value of the error counter. This is incremented by hardware when errors are logged. When this counter overflows, it stays at FFFh (no rollover). The threshold value, written by software, is (FFFh - the desired error count (the number of errors necessary in order for an interrupt to be taken)); the desired error count of 0 (a write value of FFFh) is not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::EX::MCA_MISC0_EX[Locked]) ? Read-write : Read-only.
31:24	<b>BlkPtr.</b> Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

**MSRC000\_2054 [EX Machine Check Configuration Thread 0] (MCA::EX::MCA\_CONFIG\_EX)**

Reset: 0000_0000_0000_0121h.	
Controls configuration of the associated machine check bank.	
_lthree[1:0]_core[7:0]_thread[1:0]_inst5_n[23:0]_aliasMSR; MSRC000_2054	
Bits	Description
63:41	Reserved.
40	<b>IntEn.</b> Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	<b>DeferredIntType.</b> Read-write. Reset: 0h. Specifies the type of interrupt signaled when a deferred error is logged. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[DeferredLvtOffset]). 10b = SMI trigger event. 11b = Reserved.
36:33	Reserved.
32	<b>McaXEnable.</b> Read-write. Reset: 0. Init: BIOS,1. Check: 1. 1=Software has acknowledged support for the MCAX feature set. 0=Software has not acknowledged support for the MCAX feature set. All uncorrected and fatal errors will cause an ErrorEvent packet to be generated. Deferred error interrupts are configured via Core::X86::Msr::McaIntrCfg.
31:11	Reserved.
10	<b>IntPresent.</b> Read-only, Volatile. Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::EX::MCA_CONFIG_EX[IntEn].
9	<b>McaFruTextInMca.</b> Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	<b>McaLsbInStatusSupported.</b> Read-only. Reset: 1. 1=MCA::EX::MCA_CONFIG_EX[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	<b>DeferredIntTypeSupported.</b> Read-only. Reset: 1. 1=MCA::EX::MCA_CONFIG_EX[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::EX::MCA_CONFIG_EX[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	<b>DeferredErrorLoggingSupported.</b> Read-only. Reset: 0. 1=Deferred errors are supported in this MCA bank, and the LogDeferredInMcaStat field in this register controls the logging behavior of these errors. 0=Deferred errors are not supported in this bank.
1	Reserved.
0	<b>McaX.</b> Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::EX::MCA_MISC0_EX[BlkPtr] indicates the presence of the additional MISC registers, but is not used to determine their MSR numbers. Deferred error interrupt type is specifiable by MCA bank. MCA::EX::MCA_STATUS_EX[TCC] is present.

**MSRC000\_2055 [EX IP Identification Thread 0] (MCA::EX::MCA\_IPID\_EX)**

Reset: 0005\_00B0\_0000\_0000h.

The MCA::EX::MCA\_IPID\_EX register is used by software to determine what IP type and revision is associated with the MCA bank.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst5\_n[23:0]\_aliasMSR; MSRC000\_2055

Bits	Description
63:48	<b>McaType.</b> Read-only. Reset: 0005h. The McaType of the MCA bank within this IP.
47:44	<b>InstanceIdHi.</b> Read-write. Reset: 0h. The HI value instance ID of this IP. This is initialized to a unique ID per instance of this register.
43:32	<b>HardwareID.</b> Read-only. Reset: 0B0h. The Hardware ID of the IP associated with this MCA bank.
31:0	<b>InstanceId.</b> Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _lthree0_core0_thread0_inst5_n0_aliasMSR: PSP,2080_9A00h
	Init: _lthree0_core0_thread1_inst5_n1_aliasMSR: PSP,2080_9B00h
	Init: _lthree0_core1_thread0_inst5_n2_aliasMSR: PSP,2082_9A00h
	Init: _lthree0_core1_thread1_inst5_n3_aliasMSR: PSP,2082_9B00h
	Init: _lthree0_core2_thread0_inst5_n4_aliasMSR: PSP,2084_9A00h
	Init: _lthree0_core2_thread1_inst5_n5_aliasMSR: PSP,2084_9B00h
	Init: _lthree0_core3_thread0_inst5_n6_aliasMSR: PSP,2086_9A00h
	Init: _lthree0_core3_thread1_inst5_n7_aliasMSR: PSP,2086_9B00h
	Init: _lthree1_core0_thread0_inst5_n8_aliasMSR: PSP,20C0_9A00h
	Init: _lthree1_core0_thread1_inst5_n9_aliasMSR: PSP,20C0_9B00h
	Init: _lthree1_core1_thread0_inst5_n10_aliasMSR: PSP,20C2_9A00h
	Init: _lthree1_core1_thread1_inst5_n11_aliasMSR: PSP,20C2_9B00h
	Init: _lthree1_core2_thread0_inst5_n12_aliasMSR: PSP,20C4_9A00h
	Init: _lthree1_core2_thread1_inst5_n13_aliasMSR: PSP,20C4_9B00h
	Init: _lthree1_core3_thread0_inst5_n14_aliasMSR: PSP,20C6_9A00h
	Init: _lthree1_core3_thread1_inst5_n15_aliasMSR: PSP,20C6_9B00h
	Init: _lthree1_core4_thread0_inst5_n16_aliasMSR: PSP,20C8_9A00h
	Init: _lthree1_core4_thread1_inst5_n17_aliasMSR: PSP,20C8_9B00h
	Init: _lthree1_core5_thread0_inst5_n18_aliasMSR: PSP,20CA_9A00h
	Init: _lthree1_core5_thread1_inst5_n19_aliasMSR: PSP,20CA_9B00h
	Init: _lthree1_core6_thread0_inst5_n20_aliasMSR: PSP,20CC_9A00h
	Init: _lthree1_core6_thread1_inst5_n21_aliasMSR: PSP,20CC_9B00h
	Init: _lthree1_core7_thread0_inst5_n22_aliasMSR: PSP,20CE_9A00h
	Init: _lthree1_core7_thread1_inst5_n23_aliasMSR: PSP,20CE_9B00h

**MSRC000\_2056 [EX Machine Check Syndrome Thread 0] (MCA::EX::MCA\_SYND\_EX)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::EX::MCA\_STATUS\_EX Thread 0

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst5\_n[23:0]\_aliasMSR; MSRC000\_2056

Bits	Description
63:32	<b>Syndrom.</b> Read-write, Volatile. Reset: Cold, 0000_0000h. Contains the syndrome, if any, associated with the error logged in MCA::EX::MCA_STATUS_EX. The low-order bit of the syndrome is stored in bit 0, and the syndrome has a length specified by MCA::EX::MCA_SYND_EX[Length]. The Syndrome field is only valid when MCA::EX::MCA_SYND_EX[Length] is not 0.
31:27	Reserved.
26:24	<b>ErrorPriority.</b> Read-write, Volatile. Reset: Cold, 0h. Encodes the priority of the error logged in MCA::EX::MCA_SYND_EX. 3'b000 = No error; 3'b001 = Reserved; 3'b010 = Corrected Error; 3'b011 = Deferred Error; 3'b100 = Uncorrected Error; 3'b101 = Fatal Error; all others reserved.
23:18	<b>Length.</b> Read-write, Volatile. Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::EX::MCA_SYND_EX[Syndrome]. A value of 0 indicates that there is no valid syndrome in MCA::EX::MCA_SYND_EX. For example, a syndrome length of 9 means that MCA::EX::MCA_SYND_EX[Syndrome] bits [8:0] contains a valid syndrome.
17:0	<b>ErrorInformation.</b> Read-write, Volatile. Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 51 [MCA_SYND_EX].

Table 51: MCA\_SYND\_EX

Error Type	Bits	Description
WDT	[17:0]	Reserved
PRF	[17:0]	Reserved
FRF	[17:8] [7:0]	0 FRF Index
IDRF	[17:8] [7:7] [6:0]	0 1: ALSQ, 0: AGSQ Index
PLDAG	[17:8] [7:7] [6:0]	0 1: AGSQ PLD, 0: AGSQ EPLD Index
PLDAL	[17:8] [7:7] [6:0]	0 1: ALSQ PLD, 0: ALSQ EPLD Index
CHKPTQ	[17:4] [3] [2] [1] [0]	0 Thread 1 Parity Error Thread 0 Parity Error Thread 1 Flush Parity Error Thread 0 Flush Parity Error
RETDISP	[17:2] [1] [0]	0 Thread 1 Parity Error Thread 0 Parity Error
STATQ	[17:0]	Reserved
SQ	[17:6] [5:0]	0 EXTID
BBQ	[17:3] [2] [1] [0]	0 RIP Parity Error FIP Parity Error LBF Parity Error
HWA	[17:6] [5:0]	Reserved Reserved
SPECMAP	[17:2] [1] [0]	0 Reserved Reserved
RETMAP	[17:0]	Reserved
Reserved	[17:0]	Reserved

**MSRC001\_0405 [EX Machine Check Control Mask Thread 0] (MCA::EX::MCA\_CTL\_MASK\_EX)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

Inhibit detection of an error source.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst5\_n[23:0]\_aliasMSR; MSRC001\_0405

Bits	Description
63:14	Reserved.
13	<b>RETMAP</b> . Read-write. Reset: 0. Retire Map parity error.
12	<b>SPECMAP</b> . Read-write. Reset: 0. PRN/FRN freelist parity error.
11	<b>HWA</b> . Read-write. Reset: 0. Hardware Assertion error.
10	<b>BBQ</b> . Read-write. Reset: 0. Branch buffer queue parity error.
9	<b>SQ</b> . Read-write. Reset: 0. EXTID parity error.
8	<b>STATQ</b> . Read-write. Reset: 0. Retire status queue parity error.
7	<b>RETDISP</b> . Read-write. Reset: 0. Retire dispatch queue parity error.
6	<b>CHKPTQ</b> . Read-write. Reset: 0. CHKPTQ. Checkpoint queue parity error.
5	<b>PLDAL</b> . Read-write. Reset: 0. EX payload parity error.
4	<b>PLDAG</b> . Read-write. Reset: 0. Address generator payload parity error.
3	<b>IDRF</b> . Read-write. Reset: 0. Immediate displacement register file parity error.
2	<b>FRF</b> . Read-write. Reset: 0. Flag register file parity error.
1	<b>PRF</b> . Read-write. Reset: 0. Physical register file parity error.
0	<b>WDT</b> . Read-write. Reset: 0. Watchdog Timeout error.

**MSRC000\_205E [EX Machine Check Syndrome Extended Thread 0] (MCA::EX::MCA\_SYND1\_EX)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::EX::MCA\_STATUS\_EX Thread 0

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst5\_n[23:0]\_aliasMSR; MSRC000\_205E

Bits	Description
63:0	<b>Syndrome</b> . Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::EX::MCA_SYND1_EX register stores information associated with the error in MCA::EX::MCA_STATUS_EX or MCA_DESTAT. The register is meaningful if MCA::EX::MCA_STATUS_EX[SyndV]=1. When MCA::EX::MCA_CONFIG_EX[McaFruTextInMca]=1, MCA::EX::MCA_SYND1_EX stores ASCII FruText associated with the error.

**MSRC000\_205F [EX Machine Check Syndrome Extended Thread 0] (MCA::EX::MCA\_SYND2\_EX)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::EX::MCA\_STATUS\_EX Thread 0

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst5\_n[23:0]\_aliasMSR; MSRC000\_205F

Bits	Description
63:0	<b>Syndrome</b> . Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::EX::MCA_SYND2_EX register stores information associated with the error in MCA::EX::MCA_STATUS_EX or MCA_DESTAT. The register is meaningful if MCA::EX::MCA_STATUS_EX[SyndV]=1. When MCA::EX::MCA_CONFIG_EX[McaFruTextInMca]=1, MCA::EX::MCA_SYND2_EX stores ASCII FruText associated with the error.

## 3.2.5.6 FP

**MSR0000\_0418...MSRC000\_2060 [FP Machine Check Control Thread 0] (MCA::FP::MCA\_CTL\_FP)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

0=Disables error reporting for the corresponding error. 1=Enables error reporting via machine check exception for the corresponding error. The MCA::FP::MCA\_CTL\_FP register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG\_CTL. Does not affect error detection, correction, or logging.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst6\_n[23:0]\_aliasMSRLEGACY; MSR0000\_0418

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst6\_n[23:0]\_aliasMSR; MSRC000\_2060

Bits	Description
63:8	Reserved.
7	<b>KRF.</b> Read-write. Reset: 0. Physical K mask register file (KRF) parity error.
6	<b>HWA.</b> Read-write. Reset: 0. Hardware assertion.
5	<b>SRF.</b> Read-write. Reset: 0. Status register file (SRF) parity error.
4	<b>RQ.</b> Read-write. Reset: 0. Retire queue (RQ) parity error.
3	<b>NSQ.</b> Read-write. Reset: 0. NSQ parity error.
2	<b>SCH.</b> Read-write. Reset: 0. Schedule queue parity error.
1	<b>FL.</b> Read-write. Reset: 0. Freelist (FL) parity error.
0	<b>PRF.</b> Read-write. Reset: 0. Physical register file (PRF) parity error.



**MSR0000\_0419...MSRC000\_2061 [FP Machine Check Status Thread 0] (MCA::FP::MCA\_STATUS\_FP)**

Reset: Cold,0000\_0000\_0000\_0000h.

Logs information associated with errors.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst6\_n[23:0]\_aliasMSRLEGACY; MSR0000\_0419

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst6\_n[23:0]\_aliasMSR; MSRC000\_2061

Bits	Description
63	<b>Val.</b> Reset: Cold,0. 1=A valid error has been detected. This bit should be cleared by software after the register has been read. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
62	<b>Overflow.</b> Reset: Cold,0. 1=An error was detected while the valid bit (Val) was set; at least one error was not logged. Overflow is set independently of whether the existing error is overwritten. See 3.1.3 [Machine Check Errors]. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
61	<b>UC.</b> Reset: Cold,0. 1=The error was not corrected by hardware. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
60	<b>En.</b> Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::FP::MCA_CTL_FP. This bit is a copy of bit in MCA::FP::MCA_CTL_FP for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	<b>MiscV.</b> Reset: Cold,0. 1=Valid thresholding in MCA::FP::MCA_MISC0_FP. In certain modes, MISC registers are owned by platform firmware and will RAZ when read by non-SMM code. Therefore, it is possible for MiscV=1 and the MISC register to read as all zeros. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
58	<b>AddrV.</b> Reset: Cold,0. 1=MCA::FP::MCA_ADDR_FP contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	<b>PCC.</b> Reset: Cold,0. 1=Hardware context held by the processor may have been corrupted. Continued operation of the system may have unpredictable results. The error is not recoverable or survivable, and the system should be reinitialized. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
56	<b>ErrCoreIdVal.</b> Reset: Cold,0. 1=The ErrCoreId field is valid. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
55	<b>TCC.</b> Reset: Cold,0. 1=Hardware context of the process thread to which the error was reported may have been corrupted. Continued operation of the thread may have unpredictable results. The thread must be terminated. Only meaningful when MCA::FP::MCA_STATUS_FP[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	<b>RESERV54.</b> Reset: Cold,0. MCA_STATUS Register Reserved bit. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
53	<b>SyndV.</b> Reset: Cold,0. 1=This error logged information in MCA::FP::MCA_SYND_FP. If MCA::FP::MCA_SYND_FP[ErrorPriority] is the same as the priority of the error in MCA::FP::MCA_STATUS_FP, then the information in MCA::FP::MCA_SYND_FP is associated with the error in MCA::FP::MCA_STATUS_FP. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	<b>RESERV47.</b> Reset: Cold,00h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
46	<b>CECC.</b> Reset: Cold,0. 1=The error was a correctable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
45	<b>UECC.</b> Reset: Cold,0. 1=The error was an uncorrectable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.

44	<b>Deferred.</b> Reset: Cold,0. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; an exception is deferred until the erroneous data is consumed. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
43	<b>Poison.</b> Reset: Cold,0. 1=The error was the result of attempting to consume poisoned data. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
42:41	<b>RESERV41.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
40	<b>Scrub.</b> Reset: Cold,0. 1=The error was the result of a scrub operation. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
39:38	<b>RESERV38.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
37:32	<b>ErrCoreId.</b> Reset: Cold,00h. When ErrCoreIdVal=1 this field indicates which core within the processor is associated with the error; Otherwise this field is reserved. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
31:30	<b>RESERV30.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
29:24	<b>AddrLsb.</b> Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::FP::MCA_ADDR_FP[ErrorAddr]. A value of 0 indicates that MCA::FP::MCA_ADDR_FP[63:0] contains a valid byte address. A value of 6 indicates that MCA::FP::MCA_ADDR_FP[63:6] contains a valid cache line address and that MCA::FP::MCA_ADDR_FP[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::FP::MCA_ADDR_FP[63:12] contain a valid 4KB memory page and that MCA::FP::MCA_ADDR_FP[11:0] should be ignored by error handling software. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
23:22	<b>RESERV22.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
21:16	<b>ErrorCodeExt.</b> Reset: Cold,00h. Extended Error Code. This field is used to identify the error type for root cause analysis. This field indicates which bit position in MCA::FP::MCA_CTL_FP enables error reporting for the logged error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
15:0	<b>ErrorCode.</b> Reset: Cold,0000h. Error code for this error. See 3.1.3.3 [Error Codes] for details on decoding this field. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.

Table 52: MCA\_STATUS\_FP

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
PRF	0x0	1	1	1	0	0	0
FL	0x1	1	1	1	0	0	0
SCH	0x2	1	1	1	0	0	0
NSQ	0x3	1	1	1	0	0	0
RQ	0x4	1	1	1	0	0	0
SRF	0x5	1	1	1	0	0	0
HWA	0x6	1	1	1	0	0	0
KRF	0x7	1	1	1	0	0	0

**MSR0000\_041A...MSRC000\_2062 [FP Machine Check Address Thread 0] (MCA::FP::MCA\_ADDR\_FP)**

Read-only. Reset: Cold,0000\_0000\_0000\_0000h.

MCA::FP::MCA\_ADDR\_FP stores an address and other information associated with the error in MCA::FP::MCA\_STATUS\_FP. The register is only meaningful if MCA::FP::MCA\_STATUS\_FP[Val]=1 and MCA::FP::MCA\_STATUS\_FP[AddrV]=1.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst6\_n[23:0]\_aliasMSRLEGACY; MSR0000\_041A

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst6\_n[23:0]\_aliasMSR; MSRC000\_2062

Bits	Description
63:0	<b>ErrorAddr.</b> Read-only. Reset: Cold,0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::FP::MCA_STATUS_FP.

Table 53: MCA\_ADDR\_FP

Error Type	Bits	Description
PRF	[55:0]	Reserved
FL	[55:0]	Reserved
SCH	[55:0]	Reserved
NSQ	[55:0]	Reserved
RQ	[55:0]	Reserved
SRF	[55:0]	Reserved
HWA	[55:0]	Reserved
KRF	[55:0]	Reserved

**MSR0000\_041B...MSRC000\_2063 [FP Machine Check Miscellaneous 0 Thread 0] (MCA::FP::MCA\_MISC0\_FP)**

Log miscellaneous information associated with errors.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst6\_n[23:0]\_aliasMSRLEGACY; MSR0000\_041B

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst6\_n[23:0]\_aliasMSR; MSRC000\_2063

Bits	Description
63	<b>Valid.</b> Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	<b>CntP.</b> Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	<b>Locked.</b> Reset: 0. 1=Writes to this register are ignored. This bit is set by BIOS to indicate that this register is not available for OS use. BIOS should set this bit if ThresholdIntType is set to SMI. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
60	<b>IntP.</b> Reset: 1. 1=ThresholdIntType can be used to generate interrupts. 0=ThresholdIntType and interrupt generation are not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::FP::MCA_MISC0_FP[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	<b>LvtOffset.</b> Reset: 0h. One per die. For error thresholding interrupts, specifies the address of the LVT entry in the APIC registers as follows: LVT address = (LvtOffset shifted left 4 bits) + 500h (see Core::X86::Apic::ExtendedInterruptLvtEntries). AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::FP::MCA_MISC0_FP[Locked]) ? Read-write : Read-only.
51	<b>CntEn.</b> Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::FP::MCA_MISC0_FP[Locked]) ? Read-write : Read-only.
50:49	<b>ThresholdIntType.</b> Reset: Cold,0h. Specifies the type of interrupt signaled when Ovrlw is set and IntP==1. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[ThresholdLvtOffset]). 10b = SMI trigger event. 11b = Reserved. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::FP::MCA_MISC0_FP[Locked]) ? Read-write : Read-only.
48	<b>Ovrlw.</b> Reset: Cold,0. Set by hardware when ErrCnt transitions from FFEh to FFFh. When this field is set, ErrCnt no longer increments. When this bit is set, the interrupt selected by the ThresholdIntType field is generated. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::FP::MCA_MISC0_FP[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	<b>ErrCnt.</b> Reset: Cold,000h. This is written by software to set the starting value of the error counter. This is incremented by hardware when errors are logged. When this counter overflows, it stays at FFFh (no rollover). The threshold value, written by software, is (FFFh - the desired error count (the number of errors necessary in order for an interrupt to be taken)); the desired error count of 0 (a write value of FFFh) is not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::FP::MCA_MISC0_FP[Locked]) ? Read-write : Read-only.
31:24	<b>BlkPtr.</b> Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

**MSRC000\_2064 [FP Machine Check Configuration Thread 0] (MCA::FP::MCA\_CONFIG\_FP)**

Reset: 0000_0002_0000_0121h.	
Controls configuration of the associated machine check bank.	
_lthree[1:0]_core[7:0]_thread[1:0]_inst6_n[23:0]_aliasMSR; MSRC000_2064	
Bits	Description
63:41	Reserved.
40	<b>IntEn.</b> Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	<b>DeferredIntType.</b> Read-write. Reset: 0h. Specifies the type of interrupt signaled when a deferred error is logged. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[DeferredLvtOffset]). 10b = SMI trigger event. 11b = Reserved.
36:33	Reserved.
32	<b>McaXEnable.</b> Read-write. Reset: 0. Init: BIOS,1. Check: 1. 1=Software has acknowledged support for the MCAX feature set. 0=Software has not acknowledged support for the MCAX feature set. All uncorrected and fatal errors will cause an ErrorEvent packet to be generated. Deferred error interrupts are configured via Core::X86::Msr::McaIntrCfg.
31:11	Reserved.
10	<b>IntPresent.</b> Read-only, Volatile. Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::FP::MCA_CONFIG_FP[IntEn].
9	<b>McaFruTextInMca.</b> Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	<b>McaLsbInStatusSupported.</b> Read-only. Reset: 1. 1=MCA::FP::MCA_CONFIG_FP[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	<b>DeferredIntTypeSupported.</b> Read-only. Reset: 1. 1=MCA::FP::MCA_CONFIG_FP[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::FP::MCA_CONFIG_FP[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	<b>DeferredErrorLoggingSupported.</b> Read-only. Reset: 0. 1=Deferred errors are supported in this MCA bank, and the LogDeferredInMcaStat field in this register controls the logging behavior of these errors. 0=Deferred errors are not supported in this bank.
1	Reserved.
0	<b>McaX.</b> Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::FP::MCA_MISC0_FP[BlkPtr] indicates the presence of the additional MISC registers, but is not used to determine their MSR numbers. Deferred error interrupt type is specifiable by MCA bank. MCA::FP::MCA_STATUS_FP[TCC] is present.

**MSRC000\_2065 [FP IP Identification Thread 0] (MCA::FP::MCA\_IPID\_FP)**

Reset: 0006\_00B0\_0000\_0000h.

The MCA::FP::MCA\_IPID\_FP register is used by software to determine what IP type and revision is associated with the MCA bank.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst6\_n[23:0]\_aliasMSR; MSRC000\_2065

Bits	Description
63:48	<b>McaType</b> . Read-only. Reset: 0006h. The McaType of the MCA bank within this IP.
47:44	<b>InstanceIdHi</b> . Read-write. Reset: 0h. The HI value instance ID of this IP. This is initialized to a unique ID per instance of this register.
43:32	<b>HardwareID</b> . Read-only. Reset: 0B0h. The Hardware ID of the IP associated with this MCA bank.
31:0	<b>InstanceId</b> . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _lthree0_core0_thread0_inst6_n0_aliasMSR: PSP,2080_8600h
	Init: _lthree0_core0_thread1_inst6_n1_aliasMSR: PSP,2080_8700h
	Init: _lthree0_core1_thread0_inst6_n2_aliasMSR: PSP,2082_8600h
	Init: _lthree0_core1_thread1_inst6_n3_aliasMSR: PSP,2082_8700h
	Init: _lthree0_core2_thread0_inst6_n4_aliasMSR: PSP,2084_8600h
	Init: _lthree0_core2_thread1_inst6_n5_aliasMSR: PSP,2084_8700h
	Init: _lthree0_core3_thread0_inst6_n6_aliasMSR: PSP,2086_8600h
	Init: _lthree0_core3_thread1_inst6_n7_aliasMSR: PSP,2086_8700h
	Init: _lthree1_core0_thread0_inst6_n8_aliasMSR: PSP,20C0_8600h
	Init: _lthree1_core0_thread1_inst6_n9_aliasMSR: PSP,20C0_8700h
	Init: _lthree1_core1_thread0_inst6_n10_aliasMSR: PSP,20C2_8600h
	Init: _lthree1_core1_thread1_inst6_n11_aliasMSR: PSP,20C2_8700h
	Init: _lthree1_core2_thread0_inst6_n12_aliasMSR: PSP,20C4_8600h
	Init: _lthree1_core2_thread1_inst6_n13_aliasMSR: PSP,20C4_8700h
	Init: _lthree1_core3_thread0_inst6_n14_aliasMSR: PSP,20C6_8600h
	Init: _lthree1_core3_thread1_inst6_n15_aliasMSR: PSP,20C6_8700h
	Init: _lthree1_core4_thread0_inst6_n16_aliasMSR: PSP,20C8_8600h
	Init: _lthree1_core4_thread1_inst6_n17_aliasMSR: PSP,20C8_8700h
	Init: _lthree1_core5_thread0_inst6_n18_aliasMSR: PSP,20CA_8600h
	Init: _lthree1_core5_thread1_inst6_n19_aliasMSR: PSP,20CA_8700h
	Init: _lthree1_core6_thread0_inst6_n20_aliasMSR: PSP,20CC_8600h
	Init: _lthree1_core6_thread1_inst6_n21_aliasMSR: PSP,20CC_8700h
	Init: _lthree1_core7_thread0_inst6_n22_aliasMSR: PSP,20CE_8600h
	Init: _lthree1_core7_thread1_inst6_n23_aliasMSR: PSP,20CE_8700h

**MSRC000\_2066 [FP Machine Check Syndrome Thread 0] (MCA::FP::MCA\_SYND\_FP)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::FP::MCA\_STATUS\_FP Thread 0

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst6\_n[23:0]\_aliasMSR; MSRC000\_2066

Bits	Description
63:27	Reserved.
26:24	<b>ErrorPriority</b> . Read-write, Volatile. Reset: Cold, 0h. Encodes the priority of the error logged in MCA::FP::MCA_SYND_FP. 3'b000 = No error; 3'b001 = Reserved; 3'b010 = Corrected Error; 3'b011 = Deferred Error; 3'b100 = Uncorrected Error; 3'b101 = Fatal Error; all others reserved.
23:18	<b>Length</b> . Read-write, Volatile. Reset: Cold, 00h. Specifies the length in bits of any syndromes logged. Only meaningful if the Syndrome field exists in this register.
17:0	<b>ErrorInformation</b> . Read-write, Volatile. Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 54 [MCA_SYND_FP].

Table 54: MCA\_SYND\_FP

Error Type	Bits	Description
PRF	[17:0]	Reserved
FL	[17:0]	Reserved



SCH	[17:0]	Reserved
NSQ	[17:0]	Reserved
RQ	[17:0]	Reserved
SRF	[17:0]	Reserved
HWA	[17:0]	Reserved
KRF	[17:0]	Reserved

**MSRC001\_0406 [FP Machine Check Control Mask Thread 0] (MCA::FP::MCA\_CTL\_MASK\_FP)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

Inhibit detection of an error source.

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst6\_n[23:0]\_aliasMSR; MSRC001\_0406

Bits	Description
63:8	Reserved.
7	<b>KRF.</b> Read-write. Reset: 0. Physical K mask register file (KRF) parity error.
6	<b>HWA.</b> Read-write. Reset: 0. Hardware assertion.
5	<b>SRF.</b> Read-write. Reset: 0. Status register file (SRF) parity error.
4	<b>RQ.</b> Read-write. Reset: 0. Retire queue (RQ) parity error.
3	<b>NSQ.</b> Read-write. Reset: 0. NSQ parity error.
2	<b>SCH.</b> Read-write. Reset: 0. Schedule queue parity error.
1	<b>FL.</b> Read-write. Reset: 0. Freelist (FL) parity error.
0	<b>PRF.</b> Read-write. Reset: 0. Physical register file (PRF) parity error.

**MSRC000\_206E [FP Machine Check Syndrome Extended Thread 0] (MCA::FP::MCA\_SYND1\_FP)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::FP::MCA\_STATUS\_FP Thread 0

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst6\_n[23:0]\_aliasMSR; MSRC000\_206E

Bits	Description
63:0	<b>Syndrome.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::FP::MCA_SYND1_FP register stores information associated with the error in MCA::FP::MCA_STATUS_FP or MCA_DESTAT. The register is meaningful if MCA::FP::MCA_STATUS_FP[SyndV]=1. When MCA::FP::MCA_CONFIG_FP[McaFruTextInMca]=1, MCA::FP::MCA_SYND1_FP stores ASCII FruText associated with the error.

**MSRC000\_206F [FP Machine Check Syndrome Extended Thread 0] (MCA::FP::MCA\_SYND2\_FP)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::FP::MCA\_STATUS\_FP Thread 0

\_lthree[1:0]\_core[7:0]\_thread[1:0]\_inst6\_n[23:0]\_aliasMSR; MSRC000\_206F

Bits	Description
63:0	<b>Syndrome.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::FP::MCA_SYND2_FP register stores information associated with the error in MCA::FP::MCA_STATUS_FP or MCA_DESTAT. The register is meaningful if MCA::FP::MCA_STATUS_FP[SyndV]=1. When MCA::FP::MCA_CONFIG_FP[McaFruTextInMca]=1, MCA::FP::MCA_SYND2_FP stores ASCII FruText associated with the error.

## 3.2.5.7 L3

**MSR0000\_041C...MSRC000\_20E0 [L3 Machine Check Control] (MCA::L3::MCA\_CTL\_L3)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

0=Disables error reporting for the corresponding error. 1=Enables error reporting via machine check exception for the corresponding error. The MCA::L3::MCA\_CTL\_L3 register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG\_CTL. Does not affect error detection, correction, or logging.

\_lthree[1:0]\_inst7\_n[8,0]\_aliasMSRLEGACY; MSR0000\_041C

\_lthree[1:0]\_inst8\_n[9,1]\_aliasMSRLEGACY; MSR0000\_0420

\_lthree[1:0]\_inst9\_n[10,2]\_aliasMSRLEGACY; MSR0000\_0424

\_lthree[1:0]\_inst10\_n[11,3]\_aliasMSRLEGACY; MSR0000\_0428

\_lthree[1:0]\_inst11\_n[12,4]\_aliasMSRLEGACY; MSR0000\_042C

\_lthree[1:0]\_inst12\_n[13,5]\_aliasMSRLEGACY; MSR0000\_0430

\_lthree[1:0]\_inst13\_n[14,6]\_aliasMSRLEGACY; MSR0000\_0434

\_lthree[1:0]\_inst14\_n[15,7]\_aliasMSRLEGACY; MSR0000\_0438

\_lthree[1:0]\_inst7\_n[8,0]\_aliasMSR; MSRC000\_2070

\_lthree[1:0]\_inst8\_n[9,1]\_aliasMSR; MSRC000\_2080

\_lthree[1:0]\_inst9\_n[10,2]\_aliasMSR; MSRC000\_2090

\_lthree[1:0]\_inst10\_n[11,3]\_aliasMSR; MSRC000\_20A0

\_lthree[1:0]\_inst11\_n[12,4]\_aliasMSR; MSRC000\_20B0

\_lthree[1:0]\_inst12\_n[13,5]\_aliasMSR; MSRC000\_20C0

\_lthree[1:0]\_inst13\_n[14,6]\_aliasMSR; MSRC000\_20D0

\_lthree[1:0]\_inst14\_n[15,7]\_aliasMSR; MSRC000\_20E0

Bits	Description
63:10	Reserved.
9	<b>DsmMce.</b> Read-write. Reset: 0. Machine check error initiated by DSM action
8	<b>XiWcbParityPoison.</b> Read-write. Reset: 0. Xi Wcb Parity Poison Creation Event
7	<b>Hwa.</b> Read-write. Reset: 0. L3 Hardware Assertion.
6	<b>XiVictimQueue.</b> Read-write. Reset: 0. L3 Victim Queue Data Fabric Error.
5	<b>SdpParity.</b> Read-write. Reset: 0. SDP Parity Error from XI.
4	<b>DataArray.</b> Read-write. Reset: 0. L3M Data ECC Error.
3	<b>MultiHitTag.</b> Read-write. Reset: 0. L3M Tag Multi-way-hit Error.
2	<b>Tag.</b> Read-write. Reset: 0. L3M Tag ECC Error.
1	<b>MultiHitShadowTag.</b> Read-write. Reset: 0. Shadow Tag Macro Multi-way-hit Error.
0	<b>ShadowTag.</b> Read-write. Reset: 0. Shadow Tag Macro ECC Error.



**MSR0000\_041D...MSRC000\_20E1 [L3 Machine Check Status] (MCA::L3::MCA\_STATUS\_L3)**

Reset: Cold,0000\_0000\_0000\_0000h.

Logs information associated with errors.

\_lthree[1:0]\_inst7\_n[8,0]\_aliasMSRLEGACY; MSR0000\_041D  
 \_lthree[1:0]\_inst8\_n[9,1]\_aliasMSRLEGACY; MSR0000\_0421  
 \_lthree[1:0]\_inst9\_n[10,2]\_aliasMSRLEGACY; MSR0000\_0425  
 \_lthree[1:0]\_inst10\_n[11,3]\_aliasMSRLEGACY; MSR0000\_0429  
 \_lthree[1:0]\_inst11\_n[12,4]\_aliasMSRLEGACY; MSR0000\_042D  
 \_lthree[1:0]\_inst12\_n[13,5]\_aliasMSRLEGACY; MSR0000\_0431  
 \_lthree[1:0]\_inst13\_n[14,6]\_aliasMSRLEGACY; MSR0000\_0435  
 \_lthree[1:0]\_inst14\_n[15,7]\_aliasMSRLEGACY; MSR0000\_0439  
 \_lthree[1:0]\_inst7\_n[8,0]\_aliasMSR; MSRC000\_2071  
 \_lthree[1:0]\_inst8\_n[9,1]\_aliasMSR; MSRC000\_2081  
 \_lthree[1:0]\_inst9\_n[10,2]\_aliasMSR; MSRC000\_2091  
 \_lthree[1:0]\_inst10\_n[11,3]\_aliasMSR; MSRC000\_20A1  
 \_lthree[1:0]\_inst11\_n[12,4]\_aliasMSR; MSRC000\_20B1  
 \_lthree[1:0]\_inst12\_n[13,5]\_aliasMSR; MSRC000\_20C1  
 \_lthree[1:0]\_inst13\_n[14,6]\_aliasMSR; MSRC000\_20D1  
 \_lthree[1:0]\_inst14\_n[15,7]\_aliasMSR; MSRC000\_20E1

Bits	Description
63	<b>Val.</b> Reset: Cold,0. 1=A valid error has been detected. This bit should be cleared by software after the register has been read. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
62	<b>Overflow.</b> Reset: Cold,0. 1=An error was detected while the valid bit (Val) was set; at least one error was not logged. Overflow is set independently of whether the existing error is overwritten. See 3.1.3 [Machine Check Errors]. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
61	<b>UC.</b> Reset: Cold,0. 1=The error was not corrected by hardware. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
60	<b>En.</b> Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::L3::MCA_CTL_L3. This bit is a copy of bit in MCA::L3::MCA_CTL_L3 for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	<b>MiscV.</b> Reset: Cold,0. 1=Valid thresholding in MCA::L3::MCA_MISC0_L3. In certain modes, MISC registers are owned by platform firmware and will RAZ when read by non-SMM code. Therefore, it is possible for MiscV=1 and the MISC register to read as all zeros. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
58	<b>AddrV.</b> Reset: Cold,0. 1=MCA::L3::MCA_ADDR_L3 contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	<b>PCC.</b> Reset: Cold,0. 1=Hardware context held by the processor may have been corrupted. Continued operation of the system may have unpredictable results. The error is not recoverable or survivable, and the system should be reinitialized. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
56	<b>ErrCoreIdVal.</b> Reset: Cold,0. 1=The ErrCoreId field is valid. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
55	<b>TCC.</b> Reset: Cold,0. 1=Hardware context of the process thread to which the error was reported may have been corrupted. Continued operation of the thread may have unpredictable results. The thread must be terminated. Only meaningful when MCA::L3::MCA_STATUS_L3[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	<b>RESERV54.</b> Reset: Cold,0. MCA_STATUS Register Reserved bit. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.

53	<b>SyndV.</b> Reset: Cold,0. 1=This error logged information in MCA::L3::MCA_SYND_L3. If MCA::L3::MCA_SYND_L3[ErrorPriority] is the same as the priority of the error in MCA::L3::MCA_STATUS_L3, then the information in MCA::L3::MCA_SYND_L3 is associated with the error in MCA::L3::MCA_STATUS_L3. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	<b>RESERV47.</b> Reset: Cold,00h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
46	<b>CECC.</b> Reset: Cold,0. 1=The error was a correctable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
45	<b>UECC.</b> Reset: Cold,0. 1=The error was an uncorrectable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
44	<b>Deferred.</b> Reset: Cold,0. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; an exception is deferred until the erroneous data is consumed. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
43	<b>Poison.</b> Reset: Cold,0. 1=The error was the result of attempting to consume poisoned data. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
42:41	<b>RESERV41.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
40	<b>Scrub.</b> Reset: Cold,0. 1=The error was the result of a scrub operation. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
39:38	<b>RESERV38.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
37:32	<b>ErrCoreId.</b> Reset: Cold,00h. When ErrCoreIdVal=1 this field indicates which core within the processor is associated with the error; Otherwise this field is reserved. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
31:30	<b>RESERV30.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
29:24	<b>AddrLsb.</b> Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::L3::MCA_ADDR_L3[ErrorAddr]. A value of 0 indicates that MCA::L3::MCA_ADDR_L3[63:0] contains a valid byte address. A value of 6 indicates that MCA::L3::MCA_ADDR_L3[63:6] contains a valid cache line address and that MCA::L3::MCA_ADDR_L3[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::L3::MCA_ADDR_L3[63:12] contain a valid 4KB memory page and that MCA::L3::MCA_ADDR_L3[11:0] should be ignored by error handling software. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
23:22	<b>RESERV22.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
21:16	<b>ErrorCodeExt.</b> Reset: Cold,00h. Extended Error Code. This field is used to identify the error type for root cause analysis. This field indicates which bit position in MCA::L3::MCA_CTL_L3 enables error reporting for the logged error. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
15:0	<b>ErrorCode.</b> Reset: Cold,0000h. Error code for this error. See 3.1.3.3 [Error Codes] for details on decoding this field. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.

Table 55: MCA\_STATUS\_L3

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
ShadowTag	0x0	0/1	0/1	0/1	0	0	1
MultiHitShadowTag	0x1	1	1	1	0	0	1
Tag	0x2	0/1	0/1	0/1	0	0	1
MultiHitTag	0x3	1	1	1	0	0	1
DataArray	0x4	0/1	0/1	0/1	0/1	0	1
SdpParity	0x5	1	1	1	0	0	1
XiVictimQueue	0x6	1	1	1	0	0	1
Hwa	0x7	1	1	1	0	0	1
XiWcbParityPoison	0x8	0	0	0	1	0	1
DsmMce	0x9	0/1	0/1	0/1	0/1	0	0

**MSR0000\_041E...MSRC000\_20E2 [L3 Machine Check Address] (MCA::L3::MCA\_ADDR\_L3)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

MCA::L3::MCA\_ADDR\_L3 stores an address and other information associated with the error in MCA::L3::MCA\_STATUS\_L3. The register is only meaningful if MCA::L3::MCA\_STATUS\_L3[Val]=1 and MCA::L3::MCA\_STATUS\_L3[AddrV]=1.

\_lthree[1:0]\_inst7\_n[8,0]\_aliasMSRLEGACY; MSR0000\_041E

\_lthree[1:0]\_inst8\_n[9,1]\_aliasMSRLEGACY; MSR0000\_0422

\_lthree[1:0]\_inst9\_n[10,2]\_aliasMSRLEGACY; MSR0000\_0426

\_lthree[1:0]\_inst10\_n[11,3]\_aliasMSRLEGACY; MSR0000\_042A

\_lthree[1:0]\_inst11\_n[12,4]\_aliasMSRLEGACY; MSR0000\_042E

\_lthree[1:0]\_inst12\_n[13,5]\_aliasMSRLEGACY; MSR0000\_0432

\_lthree[1:0]\_inst13\_n[14,6]\_aliasMSRLEGACY; MSR0000\_0436

\_lthree[1:0]\_inst14\_n[15,7]\_aliasMSRLEGACY; MSR0000\_043A

\_lthree[1:0]\_inst7\_n[8,0]\_aliasMSR; MSRC000\_2072

\_lthree[1:0]\_inst8\_n[9,1]\_aliasMSR; MSRC000\_2082

\_lthree[1:0]\_inst9\_n[10,2]\_aliasMSR; MSRC000\_2092

\_lthree[1:0]\_inst10\_n[11,3]\_aliasMSR; MSRC000\_20A2

\_lthree[1:0]\_inst11\_n[12,4]\_aliasMSR; MSRC000\_20B2

\_lthree[1:0]\_inst12\_n[13,5]\_aliasMSR; MSRC000\_20C2

\_lthree[1:0]\_inst13\_n[14,6]\_aliasMSR; MSRC000\_20D2

\_lthree[1:0]\_inst14\_n[15,7]\_aliasMSR; MSRC000\_20E2

**Bits Description**

63:0 **ErrorAddr.** Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h. Unless otherwise specified by an error, contains the address associated with the error logged in MCA::L3::MCA\_STATUS\_L3. For physical addresses, the most significant bit is given by Core::X86::Cpuid::LongModeInfo[PhysAddrSize].

Table 56: MCA\_ADDR\_L3

Error Type	Bits	Description
ShadowTag	[55:16] [15:0]	Reserved 16'b{7'b{Index}, 3'b{Slice}, 6'b{0}}
MultiHitShadowTag	[55:16] [15:0]	Reserved 16'b{7'b{Index}, 3'b{Slice}, 6'b{0}}
Tag	[55:21] [20:0]	Reserved 21'b{1'b{L3MIF}, 7'b{Index}, 4'b{Bank[2:0]}, 3'b{Slice}, 6'b{0}}
MultiHitTag	[55:21]	Reserved

	[21:0]	21'b{1'b{L3MIF}, 7'b{Index}, 4'b{Bank[3:0]}, 3'b{Slice}, 6'b{0}}
dataArray	[55:52] [51:0]	Reserved Physical Address
SdpParity	[55:52] [51:0]	Reserved Physical Address
XiVictimQueue	[55:52] [51:0]	Reserved Physical Address
Hwa	[63:46] [45:0]	Reserved Reserved
XiWcbParityPoison	[55:52] [51:0]	Reserved Physical Address
DsmMce	[63:0]	Reserved

**MSR0000\_041F...MSRC000\_20E3 [L3 Machine Check Miscellaneous 0] (MCA::L3::MCA\_MISC0\_L3)**

Log miscellaneous information associated with errors.

\_lthree[1:0]\_inst7\_n[8,0]\_aliasMSRLEGACY; MSR0000\_041F  
 \_lthree[1:0]\_inst8\_n[9,1]\_aliasMSRLEGACY; MSR0000\_0423  
 \_lthree[1:0]\_inst9\_n[10,2]\_aliasMSRLEGACY; MSR0000\_0427  
 \_lthree[1:0]\_inst10\_n[11,3]\_aliasMSRLEGACY; MSR0000\_042B  
 \_lthree[1:0]\_inst11\_n[12,4]\_aliasMSRLEGACY; MSR0000\_042F  
 \_lthree[1:0]\_inst12\_n[13,5]\_aliasMSRLEGACY; MSR0000\_0433  
 \_lthree[1:0]\_inst13\_n[14,6]\_aliasMSRLEGACY; MSR0000\_0437  
 \_lthree[1:0]\_inst14\_n[15,7]\_aliasMSRLEGACY; MSR0000\_043B  
 \_lthree[1:0]\_inst7\_n[8,0]\_aliasMSR; MSRC000\_2073  
 \_lthree[1:0]\_inst8\_n[9,1]\_aliasMSR; MSRC000\_2083  
 \_lthree[1:0]\_inst9\_n[10,2]\_aliasMSR; MSRC000\_2093  
 \_lthree[1:0]\_inst10\_n[11,3]\_aliasMSR; MSRC000\_20A3  
 \_lthree[1:0]\_inst11\_n[12,4]\_aliasMSR; MSRC000\_20B3  
 \_lthree[1:0]\_inst12\_n[13,5]\_aliasMSR; MSRC000\_20C3  
 \_lthree[1:0]\_inst13\_n[14,6]\_aliasMSR; MSRC000\_20D3  
 \_lthree[1:0]\_inst14\_n[15,7]\_aliasMSR; MSRC000\_20E3

Bits	Description
63	<b>Valid.</b> Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	<b>CntP.</b> Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	<b>Locked.</b> Reset: 0. 1=Writes to this register are ignored. This bit is set by BIOS to indicate that this register is not available for OS use. BIOS should set this bit if ThresholdIntType is set to SMI. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
60	<b>IntP.</b> Reset: 1. 1=ThresholdIntType can be used to generate interrupts. 0=ThresholdIntType and interrupt generation are not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::L3::MCA_MISC0_L3[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	<b>LvtOffset.</b> Reset: 0h. One per die. For error thresholding interrupts, specifies the address of the LVT entry in the APIC registers as follows: LVT address = (LvtOffset shifted left 4 bits) + 500h (see Core::X86::Apic::ExtendedInterruptLvtEntries). AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::L3::MCA_MISC0_L3[Locked]) ? Read-write : Read-only.
51	<b>CntEn.</b> Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::L3::MCA_MISC0_L3[Locked]) ? Read-write : Read-only.
50:49	<b>ThresholdIntType.</b> Reset: Cold,0h. Specifies the type of interrupt signaled when Ovrflw is set and IntP==1. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[ThresholdLvtOffset]). 10b = SMI trigger event. 11b = Reserved. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::L3::MCA_MISC0_L3[Locked]) ? Read-write : Read-only.
48	<b>Ovrflw.</b> Reset: Cold,0. Set by hardware when ErrCnt transitions from FFEh to FFFh. When this field is set, ErrCnt no longer increments. When this bit is set, the interrupt selected by the ThresholdIntType field is generated. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::L3::MCA_MISC0_L3[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	<b>ErrCnt.</b> Reset: Cold,000h. This is written by software to set the starting value of the error counter. This is incremented by hardware when errors are logged. When this counter overflows, it stays at FFFh (no rollover). The threshold value, written by software, is (FFFh - the desired error count (the number of errors necessary in order for an interrupt to be taken)); the desired error count of 0 (a write value of FFFh) is not supported.

	AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::L3::MCA_MISC0_L3[Locked]) ? Read-write : Read-only.
31:24	<b>BlkPtr.</b> Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

**MSRC000\_20[7...E]4 [L3 Machine Check Configuration] (MCA::L3::MCA\_CONFIG\_L3)**

Reset: 0000\_0000\_0000\_0125h.

Controls configuration of the associated machine check bank.

\_lthree[1:0]\_inst7\_n[8,0]\_aliasMSR; MSRC000\_2074

\_lthree[1:0]\_inst8\_n[9,1]\_aliasMSR; MSRC000\_2084

\_lthree[1:0]\_inst9\_n[10,2]\_aliasMSR; MSRC000\_2094

\_lthree[1:0]\_inst10\_n[11,3]\_aliasMSR; MSRC000\_20A4

\_lthree[1:0]\_inst11\_n[12,4]\_aliasMSR; MSRC000\_20B4

\_lthree[1:0]\_inst12\_n[13,5]\_aliasMSR; MSRC000\_20C4

\_lthree[1:0]\_inst13\_n[14,6]\_aliasMSR; MSRC000\_20D4

\_lthree[1:0]\_inst14\_n[15,7]\_aliasMSR; MSRC000\_20E4

Bits	Description
63:41	Reserved.
40	<b>IntEn.</b> Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	<b>DeferredIntType.</b> Read-write. Reset: 0h. Specifies the type of interrupt signaled when a deferred error is logged. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[DeferredLvtOffset]). 10b = SMI trigger event. 11b = Reserved.
36:35	Reserved.
34	<b>LogDeferredInMcaStat.</b> Read-write. Reset: 0. Init: BIOS,1. 1=Log deferred errors in MCA::L3::MCA_STATUS_L3 and MCA::L3::MCA_ADDR_L3 in addition to MCA::L3::MCA_DESTAT_L3 and MCA::L3::MCA_DEADDR_L3. 0=Only log deferred errors in MCA::L3::MCA_DESTAT_L3 and MCA::L3::MCA_DEADDR_L3. This bit does not affect logging of deferred errors in MCA::L3::MCA_SYND_L3, MCA::L3::MCA_MISC0_L3.
33	Reserved.
32	<b>McaXEnable.</b> Read-write. Reset: 0. Init: BIOS,1. Check: 1. 1=Software has acknowledged support for the MCAX feature set. 0=Software has not acknowledged support for the MCAX feature set. All uncorrected and fatal errors will cause an ErrorEvent packet to be generated. Deferred error interrupts are configured via Core::X86::Msr::McaIntrCfg.
31:11	Reserved.
10	<b>IntPresent.</b> Read-only, Volatile. Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::L3::MCA_CONFIG_L3[IntEn].
9	<b>McaFruTextInMca.</b> Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	<b>McaLsbInStatusSupported.</b> Read-only. Reset: 1. 1=MCA::L3::MCA_CONFIG_L3[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	<b>DeferredIntTypeSupported.</b> Read-only. Reset: 1. 1=MCA::L3::MCA_CONFIG_L3[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::L3::MCA_CONFIG_L3[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	<b>DeferredErrorLoggingSupported.</b> Read-only. Reset: 1. 1=Deferred errors are supported in this MCA bank, and MCA::L3::MCA_CONFIG_L3[LogDeferredInMcaStat] controls the logging behavior of these errors. MCA::L3::MCA_DESTAT_L3 and MCA::L3::MCA_DEADDR_L3 are supported in this MCA bank. 0=Deferred errors are not supported in this bank.
1	Reserved.
0	<b>McaX.</b> Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::L3::MCA_MISC0_L3[BlkPtr] indicates the presence of the additional MISC registers, but is not used to determine their MSR numbers. Deferred error interrupt type is specifiable by MCA bank. MCA::L3::MCA_STATUS_L3[TCC] is present.



**MSRC000\_20[7...E]5 [L3 IP Identification] (MCA::L3::MCA\_IPID\_L3)**

Reset: 0007\_00B0\_0000\_0000h.

The MCA::L3::MCA\_IPID\_L3 register is used by software to determine what IP type and revision is associated with the MCA bank.

\_lthree[1:0]\_inst7\_n[8,0]\_aliasMSR; MSRC000\_2075

\_lthree[1:0]\_inst8\_n[9,1]\_aliasMSR; MSRC000\_2085

\_lthree[1:0]\_inst9\_n[10,2]\_aliasMSR; MSRC000\_2095

\_lthree[1:0]\_inst10\_n[11,3]\_aliasMSR; MSRC000\_20A5

\_lthree[1:0]\_inst11\_n[12,4]\_aliasMSR; MSRC000\_20B5

\_lthree[1:0]\_inst12\_n[13,5]\_aliasMSR; MSRC000\_20C5

\_lthree[1:0]\_inst13\_n[14,6]\_aliasMSR; MSRC000\_20D5

\_lthree[1:0]\_inst14\_n[15,7]\_aliasMSR; MSRC000\_20E5

Bits	Description
63:48	<b>McaType</b> . Read-only. Reset: 0007h. The McaType of the MCA bank within this IP.
47:44	<b>InstanceIdHi</b> . Read-write. Reset: 0h. The HI value instance ID of this IP. This is initialized to a unique ID per instance of this register.
43:32	<b>HardwareID</b> . Read-only. Reset: 0B0h. The Hardware ID of the IP associated with this MCA bank.
31:0	<b>InstanceId</b> . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _lthree0_inst7_n0_aliasMSR: PSP,20B4_0000h
	Init: _lthree0_inst8_n1_aliasMSR: PSP,20B4_1000h
	Init: _lthree0_inst9_n2_aliasMSR: PSP,20B4_2000h
	Init: _lthree0_inst10_n3_aliasMSR: PSP,20B4_3000h
	Init: _lthree0_inst11_n4_aliasMSR: PSP,20B4_4000h
	Init: _lthree0_inst12_n5_aliasMSR: PSP,20B4_5000h
	Init: _lthree0_inst13_n6_aliasMSR: PSP,20B4_6000h
	Init: _lthree0_inst14_n7_aliasMSR: PSP,20B4_7000h
	Init: _lthree1_inst7_n8_aliasMSR: PSP,20F4_0000h
	Init: _lthree1_inst8_n9_aliasMSR: PSP,20F4_1000h
	Init: _lthree1_inst9_n10_aliasMSR: PSP,20F4_2000h
	Init: _lthree1_inst10_n11_aliasMSR: PSP,20F4_3000h
	Init: _lthree1_inst11_n12_aliasMSR: PSP,20F4_4000h
	Init: _lthree1_inst12_n13_aliasMSR: PSP,20F4_5000h
	Init: _lthree1_inst13_n14_aliasMSR: PSP,20F4_6000h
	Init: _lthree1_inst14_n15_aliasMSR: PSP,20F4_7000h



**MSRC000\_20[7...E]6 [L3 Machine Check Syndrome] (MCA::L3::MCA\_SYND\_L3)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::L3::MCA\_STATUS\_L3 Thread 0

\_lthree[1:0]\_inst7\_n[8,0]\_aliasMSR; MSRC000\_2076

\_lthree[1:0]\_inst8\_n[9,1]\_aliasMSR; MSRC000\_2086

\_lthree[1:0]\_inst9\_n[10,2]\_aliasMSR; MSRC000\_2096

\_lthree[1:0]\_inst10\_n[11,3]\_aliasMSR; MSRC000\_20A6

\_lthree[1:0]\_inst11\_n[12,4]\_aliasMSR; MSRC000\_20B6

\_lthree[1:0]\_inst12\_n[13,5]\_aliasMSR; MSRC000\_20C6

\_lthree[1:0]\_inst13\_n[14,6]\_aliasMSR; MSRC000\_20D6

\_lthree[1:0]\_inst14\_n[15,7]\_aliasMSR; MSRC000\_20E6

Bits	Description
63:32	<b>Syndrone.</b> Read-write, Volatile. Reset: Cold, 0000_0000h. Contains the syndrome, if any, associated with the error logged in MCA::L3::MCA_STATUS_L3. The low-order bit of the syndrome is stored in bit 0, and the syndrome has a length specified by MCA::L3::MCA_SYND_L3[Length]. The Syndrome field is only valid when MCA::L3::MCA_SYND_L3[Length] is not 0.
31:27	Reserved.
26:24	<b>ErrorPriority.</b> Read-write, Volatile. Reset: Cold, 0h. Encodes the priority of the error logged in MCA::L3::MCA_SYND_L3. 3'b000 = No error; 3'b001 = Reserved; 3'b010 = Corrected Error; 3'b011 = Deferred Error; 3'b100 = Uncorrected Error; 3'b101 = Fatal Error; all others reserved.
23:18	<b>Length.</b> Read-write, Volatile. Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::L3::MCA_SYND_L3[Syndrome]. A value of 0 indicates that there is no valid syndrome in MCA::L3::MCA_SYND_L3. For example, a syndrome length of 9 means that MCA::L3::MCA_SYND_L3[Syndrome] bits [8:0] contains a valid syndrome.
17:0	<b>ErrorInformation.</b> Read-write, Volatile. Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 57 [MCA_SYND_L3].

Table 57: MCA\_SYND\_L3

Error Type	Bits	Description
ShadowTag	[17:12] [11:8] [7:4] [3:0]	Reserved Pack Reserved Way
MultiHitShadowTag	[17:12] [11:8] [7:0]	Reserved Pack Reserved
Tag	[17:14] [13] [12] [11:8] [7:4] [3:0]	Reserved Reserved Reserved Bank. Reserved Way
MultiHitTag	[17:0]	Reserved
DataArray	[17:14] [13] [12] [11] [10:8] [7:4] [3:0]	Reserved Reserved Reserved Reserved Bank[2:0] Reserved Way
SdpParity	[17:0]	Reserved

XiVictimQueue	[17:0]	Reserved
Hwa	[17:0]	Reserved
XiWcbParityPoison	[17:0]	Reserved
DsmMce	[17:0]	Reserved

**MSRC000\_20[7...E]8 [L3 Machine Check Deferred Error Status] (MCA::L3::MCA\_DESTAT\_L3)**

Reset: Cold,0000\_0000\_0000\_0000h.

Holds status information for the first deferred error seen in this bank.

\_lthree[1:0]\_inst7\_n[8,0]\_aliasMSR; MSRC000\_2078

\_lthree[1:0]\_inst8\_n[9,1]\_aliasMSR; MSRC000\_2088

\_lthree[1:0]\_inst9\_n[10,2]\_aliasMSR; MSRC000\_2098

\_lthree[1:0]\_inst10\_n[11,3]\_aliasMSR; MSRC000\_20A8

\_lthree[1:0]\_inst11\_n[12,4]\_aliasMSR; MSRC000\_20B8

\_lthree[1:0]\_inst12\_n[13,5]\_aliasMSR; MSRC000\_20C8

\_lthree[1:0]\_inst13\_n[14,6]\_aliasMSR; MSRC000\_20D8

\_lthree[1:0]\_inst14\_n[15,7]\_aliasMSR; MSRC000\_20E8

Bits	Description
63	<b>Val.</b> Read-write, Volatile. Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	<b>Overflow.</b> Read-write, Volatile. Reset: Cold,0. 1=An error was detected while the valid bit (Val) was set; at least one error was not logged. Overflow is set independently of whether the existing error is overwritten. (See the section on error overwrite priorities.)
61:59	<b>RESERV4.</b> Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	<b>AddrV.</b> Read-write, Volatile. Reset: Cold,0. 1=MCA::L3::MCA_DEADDR_L3 contains address information associated with the error.
57:54	<b>RESERV3.</b> Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	<b>SyndV.</b> Read-write, Volatile. Reset: Cold,0. 1=This error logged information in MCA::L3::MCA_SYND_L3. If MCA::L3::MCA_SYND_L3[ErrorPriority] is the same as the priority of the error in MCA::L3::MCA_STATUS_L3, then the information in MCA::L3::MCA_SYND_L3 is associated with the error in MCA::L3::MCA_DESTAT_L3.
52:45	<b>RESERV2.</b> Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	<b>Deferred.</b> Read-write, Volatile. Reset: Cold,0. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; poison is created and an exception is deferred until the poison data is consumed.
43:30	<b>RESERV1.</b> Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	<b>AddrLsb.</b> Read-write, Volatile. Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::L3::MCA_ADDR_L3[ErrorAddr]. A value of 0 indicates that MCA::L3::MCA_ADDR_L3[63:0] contains a valid byte address. A value of 6 indicates that MCA::L3::MCA_ADDR_L3[63:6] contains a valid cache line address and that MCA::L3::MCA_ADDR_L3[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::L3::MCA_ADDR_L3[63:12] contain a valid 4KB memory page and that MCA::L3::MCA_ADDR_L3[11:0] should be ignored by error handling software.
23:22	<b>RESERV0.</b> Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	<b>ErrorCodeExt.</b> Read-write, Volatile. Reset: Cold,00h. Logs an extended error code when an error is detected. This model-specific field is used in conjunction with ErrorCode to identify the error sub-type for root cause analysis.
15:0	<b>ErrorCode.</b> Read-write, Volatile. Reset: Cold,0000h. Error code for this error.

**MSRC000\_20[7...E]9 [L3 Deferred Error Address] (MCA::L3::MCA\_DEADDR\_L3)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

The MCA::L3::MCA\_DEADDR\_L3 register stores the address associated with the error in MCA::L3::MCA\_DESTAT\_L3. The register is only meaningful if MCA::L3::MCA\_DESTAT\_L3[Val]=1 and MCA::L3::MCA\_DESTAT\_L3[AddrV]=1. The lowest valid bit of the address is defined by MCA::L3::MCA\_DESTAT\_L3[AddrLsb].

\_lthree[1:0]\_inst7\_n[8,0]\_aliasMSR; MSRC000\_2079

\_lthree[1:0]\_inst8\_n[9,1]\_aliasMSR; MSRC000\_2089

\_lthree[1:0]\_inst9\_n[10,2]\_aliasMSR; MSRC000\_2099

\_lthree[1:0]\_inst10\_n[11,3]\_aliasMSR; MSRC000\_20A9

\_lthree[1:0]\_inst11\_n[12,4]\_aliasMSR; MSRC000\_20B9

\_lthree[1:0]\_inst12\_n[13,5]\_aliasMSR; MSRC000\_20C9

\_lthree[1:0]\_inst13\_n[14,6]\_aliasMSR; MSRC000\_20D9

\_lthree[1:0]\_inst14\_n[15,7]\_aliasMSR; MSRC000\_20E9

Bits	Description
63:0	<b>ErrorAddr.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::L3::MCA_DESTAT_L3. The lowest-order valid bit of the address is specified in MCA::L3::MCA_DESTAT_L3[AddrLsb].

**MSRC001\_040[7...E] [L3 Machine Check Control Mask] (MCA::L3::MCA\_CTL\_MASK\_L3)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

Inhibit detection of an error source.

\_lthree[1:0]\_inst7\_n[8,0]\_aliasMSR; MSRC001\_0407

\_lthree[1:0]\_inst8\_n[9,1]\_aliasMSR; MSRC001\_0408

\_lthree[1:0]\_inst9\_n[10,2]\_aliasMSR; MSRC001\_0409

\_lthree[1:0]\_inst10\_n[11,3]\_aliasMSR; MSRC001\_040A

\_lthree[1:0]\_inst11\_n[12,4]\_aliasMSR; MSRC001\_040B

\_lthree[1:0]\_inst12\_n[13,5]\_aliasMSR; MSRC001\_040C

\_lthree[1:0]\_inst13\_n[14,6]\_aliasMSR; MSRC001\_040D

\_lthree[1:0]\_inst14\_n[15,7]\_aliasMSR; MSRC001\_040E

Bits	Description
63:10	Reserved.
9	<b>DsmMce.</b> Read-write. Reset: 0. Machine check error initiated by DSM action
8	<b>XiWcbParityPoison.</b> Read-write. Reset: 0. Xi Wcb Parity Poison Creation Event
7	<b>Hwa.</b> Read-write. Reset: 0. L3 Hardware Assertion.
6	<b>XiVictimQueue.</b> Read-write. Reset: 0. L3 Victim Queue Data Fabric Error.
5	<b>SdpParity.</b> Read-write. Reset: 0. SDP Parity Error from XI.
4	<b>DataArray.</b> Read-write. Reset: 0. L3M Data ECC Error.
3	<b>MultiHitTag.</b> Read-write. Reset: 0. L3M Tag Multi-way-hit Error.
2	<b>Tag.</b> Read-write. Reset: 0. L3M Tag ECC Error.
1	<b>MultiHitShadowTag.</b> Read-write. Reset: 0. Shadow Tag Macro Multi-way-hit Error.
0	<b>ShadowTag.</b> Read-write. Reset: 0. Shadow Tag Macro ECC Error.

**MSRC000\_20[7...E]E [L3 Machine Check Syndrome Extended] (MCA::L3::MCA\_SYND1\_L3)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::L3::MCA\_STATUS\_L3 Thread 0

\_lthree[1:0]\_inst7\_n[8,0]\_aliasMSR; MSRC000\_207E\_lthree[1:0]\_inst8\_n[9,1]\_aliasMSR; MSRC000\_208E\_lthree[1:0]\_inst9\_n[10,2]\_aliasMSR; MSRC000\_209E\_lthree[1:0]\_inst10\_n[11,3]\_aliasMSR; MSRC000\_20AE\_lthree[1:0]\_inst11\_n[12,4]\_aliasMSR; MSRC000\_20BE\_lthree[1:0]\_inst12\_n[13,5]\_aliasMSR; MSRC000\_20CE\_lthree[1:0]\_inst13\_n[14,6]\_aliasMSR; MSRC000\_20DE\_lthree[1:0]\_inst14\_n[15,7]\_aliasMSR; MSRC000\_20EE

Bits	Description
63:0	<b>Syndrome.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::L3::MCA_SYND1_L3 register stores information associated with the error in MCA::L3::MCA_STATUS_L3 or MCA_DESTAT. The register is meaningful if MCA::L3::MCA_STATUS_L3[SyndV]=1. When MCA::L3::MCA_CONFIG_L3[McaFruTextInMca]=1, MCA::L3::MCA_SYND1_L3 stores ASCII FruText associated with the error.

**MSRC000\_20[7...E]F [L3 Machine Check Syndrome Extended] (MCA::L3::MCA\_SYND2\_L3)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::L3::MCA\_STATUS\_L3 Thread 0

\_lthree[1:0]\_inst7\_n[8,0]\_aliasMSR; MSRC000\_207F\_lthree[1:0]\_inst8\_n[9,1]\_aliasMSR; MSRC000\_208F\_lthree[1:0]\_inst9\_n[10,2]\_aliasMSR; MSRC000\_209F\_lthree[1:0]\_inst10\_n[11,3]\_aliasMSR; MSRC000\_20AF\_lthree[1:0]\_inst11\_n[12,4]\_aliasMSR; MSRC000\_20BF\_lthree[1:0]\_inst12\_n[13,5]\_aliasMSR; MSRC000\_20CF\_lthree[1:0]\_inst13\_n[14,6]\_aliasMSR; MSRC000\_20DF\_lthree[1:0]\_inst14\_n[15,7]\_aliasMSR; MSRC000\_20EF

Bits	Description
63:0	<b>Syndrome.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::L3::MCA_SYND2_L3 register stores information associated with the error in MCA::L3::MCA_STATUS_L3 or MCA_DESTAT. The register is meaningful if MCA::L3::MCA_STATUS_L3[SyndV]=1. When MCA::L3::MCA_CONFIG_L3[McaFruTextInMca]=1, MCA::L3::MCA_SYND2_L3 stores ASCII FruText associated with the error.

## 3.2.5.8 CS

<b>MSR0000_044C...MSRC000_2160 [CS Machine Check Control] (MCA::CS::MCA_CTL_CS)</b>	
Read-write. Reset: 0000_0000_0000_0000h.	
0=Disables error reporting for the corresponding error. 1=Enables error reporting via machine check exception for the corresponding error. The MCA::CS::MCA_CTL_CS register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG_CTL. Does not affect error detection, correction, or logging.	
_instCS0_n0_aliasMSRLEGACY; MSR0000_044C	
_instCS1_n1_aliasMSRLEGACY; MSR0000_0450	
_instCS2_n2_aliasMSRLEGACY; MSR0000_0454	
_instCS3_n3_aliasMSRLEGACY; MSR0000_0458	
_instCS0_n0_aliasMSR; MSRC000_2130	
_instCS1_n1_aliasMSR; MSRC000_2140	
_instCS2_n2_aliasMSR; MSRC000_2150	
_instCS3_n3_aliasMSR; MSRC000_2160	
Bits	Description
63:21	Reserved.
20	<b>ST_TXN_ERR.</b> Read-write. Reset: 0. Shadow Tag Transaction Error: An illegal Shadow Tag access occurred.
19	<b>ST_ECC_ERR.</b> Read-write. Reset: 0. Shadow Tag ECC Error: An ECC error occurred on a shadow tag array access.
18	<b>ST_PRT_ERR.</b> Read-write. Reset: 0. Shadow Tag Array Protocol Error: Indicates a Cache Coherence Issue.
17	<b>HWA.</b> Read-write. Reset: 0. Hardware Assert Error.
16	<b>FTI_ND_SEC_VIOL.</b> Read-write. Reset: 0. Security Violation: A security violation was detected on an incoming request from the transport layer on the request no data channel.
15	<b>FTI_ND_ADDR_VIOL.</b> Read-write. Reset: 0. Address Violation: An address violation was detected on an incoming request from the transport layer on the request no data channel.
14	<b>FTI_ND_ILL_REQ.</b> Read-write. Reset: 0. Illegal Request: An illegal request was received from the transport layer on the request no data channel.
13	<b>CNTR_UNFL.</b> Read-write. Reset: 0. Counter underflow error.
12	<b>CNTR_OVFL.</b> Read-write. Reset: 0. Counter overflow error.
11	<b>SDP_UNEXP_RETRY.</b> Read-write. Reset: 0. SDP read response had an unexpected RETRY error.
10	<b>SPF_ECC_ERR.</b> Read-write. Reset: 0. Probe Filter ECC Error: An ECC error occurred on a probe filter access.
9	<b>SPF_PRT_ERR.</b> Read-write. Reset: 0. Probe Filter Protocol Error: Indicates a Cache Coherence Issue.
8	<b>SDP_RSP_NO_MTCH.</b> Read-write. Reset: 0. SDP read response had no match in the CS queue.
7	<b>ATM_PAR_ERR.</b> Read-write. Reset: 0. Atomic Request Parity Error: Parity error on read of an atomic transaction.
6	<b>SDP_PAR_ERR.</b> Read-write. Reset: 0. Read Response Parity Error: Parity error on incoming read response data.
5	<b>FTI_PAR_ERR.</b> Read-write. Reset: 0. Request or Probe Parity Error: Parity error on incoming request or probe response data.
4	<b>FTI_RSP_NO_MTCH.</b> Read-write. Reset: 0. Unexpected Response: A response was received from the transport layer which does not match any request.
3	<b>FTI_ILL_RSP.</b> Read-write. Reset: 0. Illegal Response: An illegal response was received from the transport layer.
2	<b>FTI_SEC_VIOL.</b> Read-write. Reset: 0. Security Violation: A security violation was detected on an incoming request from the transport layer on the primary request channel.
1	<b>FTI_ADDR_VIOL.</b> Read-write. Reset: 0. Address Violation: An address violation was detected on an incoming request from the transport layer on the primary request channel.
0	<b>FTI_ILL_REQ.</b> Read-write. Reset: 0. Illegal Request: An illegal request was received from the transport layer on the primary request channel.

**MSR0000\_044D...MSRC000\_2161 [CS Machine Check Status] (MCA::CS::MCA\_STATUS\_CS)**

Reset: Cold,0000\_0000\_0000\_0000h.

Logs information associated with errors.

\_instCS0\_n0\_aliasMSRLEGACY; MSR0000\_044D\_instCS1\_n1\_aliasMSRLEGACY; MSR0000\_0451\_instCS2\_n2\_aliasMSRLEGACY; MSR0000\_0455\_instCS3\_n3\_aliasMSRLEGACY; MSR0000\_0459\_instCS0\_n0\_aliasMSR; MSRC000\_2131\_instCS1\_n1\_aliasMSR; MSRC000\_2141\_instCS2\_n2\_aliasMSR; MSRC000\_2151\_instCS3\_n3\_aliasMSR; MSRC000\_2161

Bits	Description
63	<b>Val.</b> Reset: Cold,0. 1=A valid error has been detected. This bit should be cleared by software after the register has been read. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
62	<b>Overflow.</b> Reset: Cold,0. 1=An error was detected while the valid bit (Val) was set; at least one error was not logged. Overflow is set independently of whether the existing error is overwritten. See 3.1.3 [Machine Check Errors]. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
61	<b>UC.</b> Reset: Cold,0. 1=The error was not corrected by hardware. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
60	<b>En.</b> Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::CS::MCA_CTL_CS. This bit is a copy of bit in MCA::CS::MCA_CTL_CS for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	<b>MiscV.</b> Reset: Cold,0. 1=Valid thresholding in MCA::CS::MCA_MISC0_CS. In certain modes, MISC registers are owned by platform firmware and will RAZ when read by non-SMM code. Therefore, it is possible for MiscV=1 and the MISC register to read as all zeros. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
58	<b>AddrV.</b> Reset: Cold,0. 1=MCA::CS::MCA_ADDR_CS contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	<b>PCC.</b> Reset: Cold,0. 1=Hardware context held by the processor may have been corrupted. Continued operation of the system may have unpredictable results. The error is not recoverable or survivable, and the system should be reinitialized. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
56	<b>ErrCoreIdVal.</b> Reset: Cold,0. 1=The ErrCoreId field is valid. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
55	<b>TCC.</b> Reset: Cold,0. 1=Hardware context of the process thread to which the error was reported may have been corrupted. Continued operation of the thread may have unpredictable results. The thread must be terminated. Only meaningful when MCA::CS::MCA_STATUS_CS[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	<b>RESERV54.</b> Reset: Cold,0. MCA_STATUS Register Reserved bit. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
53	<b>SyndV.</b> Reset: Cold,0. 1=This error logged information in MCA::CS::MCA_SYND_CS. If MCA::CS::MCA_SYND_CS[ErrorPriority] is the same as the priority of the error in MCA::CS::MCA_STATUS_CS, then the information in MCA::CS::MCA_SYND_CS is associated with the error in MCA::CS::MCA_STATUS_CS. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	<b>RESERV47.</b> Reset: Cold,00h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
46	<b>CECC.</b> Reset: Cold,0. 1=The error was a correctable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor.



	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
45	<b>UECC</b> . Reset: Cold,0. 1=The error was an uncorrectable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
44	<b>Deferred</b> . Reset: Cold,0. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; an exception is deferred until the erroneous data is consumed.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
43	<b>Poison</b> . Reset: Cold,0. 1=The error was the result of attempting to consume poisoned data.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
42:41	<b>RESERV41</b> . Reset: Cold,0h. MCA_STATUS Register Reserved bits.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
40	<b>Scrub</b> . Reset: Cold,0. 1=The error was the result of a scrub operation.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
39:38	<b>RESERV38</b> . Reset: Cold,0h. MCA_STATUS Register Reserved bits.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
37:32	<b>ErrCoreId</b> . Reset: Cold,00h. When ErrCoreIdVal=1 this field indicates which core within the processor is associated with the error; Otherwise this field is reserved.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
31:30	<b>RESERV30</b> . Reset: Cold,0h. MCA_STATUS Register Reserved bits.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
29:24	<b>AddrLsb</b> . Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::CS::MCA_ADDR_CS[ErrorAddr]. A value of 0 indicates that MCA::CS::MCA_ADDR_CS[63:0] contains a valid byte address. A value of 6 indicates that MCA::CS::MCA_ADDR_CS[63:6] contains a valid cache line address and that MCA::CS::MCA_ADDR_CS[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::CS::MCA_ADDR_CS[63:12] contain a valid 4KB memory page and that MCA::CS::MCA_ADDR_CS[11:0] should be ignored by error handling software.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
23:22	<b>RESERV22</b> . Reset: Cold,0h. MCA_STATUS Register Reserved bits.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
21:16	<b>ErrorCodeExt</b> . Reset: Cold,00h. Extended Error Code. This field is used to identify the error type for root cause analysis. This field indicates which bit position in MCA::CS::MCA_CTL_CS enables error reporting for the logged error.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
15:0	<b>ErrorCode</b> . Reset: Cold,0000h. Error code for this error. See 3.1.3.3 [Error Codes] for details on decoding this field.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.

Table 58: MCA\_STATUS\_CS

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
FTI_ILL_REQ	0x0	0	0	0	1	0	1
FTI_ADDR_VIOL	0x1	0	0	0	1	0	1
FTI_SEC_VIOL	0x2	0	0	0	1	0	1
FTI_ILL_RSP	0x3	1	1	1	0	0	0

FTI_RSP_N O_MTCH	0x4	1	1	1	0	0	0
FTI_PAR_E RR	0x5	0	0	0	1	0	1
SDP_PAR_E RR	0x6	0	0	0	1	0	1
ATM_PAR_ ERR	0x7	0	0	0	1	0	1
SDP_RSP_N O_MTCH	0x8	1	1	1	0	0	0
SPF_PRT_E RR	0x9	1	1	1	0	0	0
SPF_ECC_E RR	0xa	0/1	0/1	0/1	0	0	1
SDP_UNEX P_RETRY	0xb	1	1	1	0	0	1
CNTR_OVF L	0xc	1	1	1	0	0	0
CNTR_UNF L	0xd	1	1	1	0	0	0
FTI_ND_IL L_REQ	0xe	0	0	0	1	0	1
FTI_ND_A DDR_VIOL	0xf	0	0	0	1	0	1
FTI_ND_SE C_VIOL	0x10	0	0	0	1	0	1
HWA	0x11	1	1	1	0	0	0
ST_PRT_ER R	0x12	1	1	1	0	0	0
ST_ECC_E RR	0x13	0/1	0/1	0/1	0	0	0
ST_TXN_E RR	0x14	1	1	1	0	0	0

**MSR0000\_044E...MSRC000\_2162 [CS Machine Check Address] (MCA::CS::MCA\_ADDR\_CS)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

MCA::CS::MCA\_ADDR\_CS stores an address and other information associated with the error in MCA::CS::MCA\_STATUS\_CS. The register is only meaningful if MCA::CS::MCA\_STATUS\_CS[Val]=1 and MCA::CS::MCA\_STATUS\_CS[AddrV]=1.

\_instCS0\_n0\_aliasMSRLEGACY; MSR0000\_044E

\_instCS1\_n1\_aliasMSRLEGACY; MSR0000\_0452

\_instCS2\_n2\_aliasMSRLEGACY; MSR0000\_0456

\_instCS3\_n3\_aliasMSRLEGACY; MSR0000\_045A

\_instCS0\_n0\_aliasMSR; MSRC000\_2132

\_instCS1\_n1\_aliasMSR; MSRC000\_2142

\_instCS2\_n2\_aliasMSR; MSRC000\_2152

\_instCS3\_n3\_aliasMSR; MSRC000\_2162

Bits	Description
63:0	<b>ErrorAddr.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. Unless otherwise specified by an error, contains the address associated with the error logged in MCA::CS::MCA_STATUS_CS. For physical addresses, the most significant bit is given by Core::X86::Cpuid::LongModeInfo[PhysAddrSize].



Table 59: MCA\_ADDR\_CS

Error Type	Bits	Description
FTI_ILL_REQ	[47:2]	Address
FTI_ADDR_VIOL	[47:2]	Address
FTI_SEC_VIOL	[47:2]	Address
FTI_ILL_RSP	[63:0]	Reserved
FTI_RSP_NO_MTCH	[63:0]	Reserved
FTI_PAR_ERR	[47:2]	Address
SDP_PAR_ERR	[47:2]	Address
ATM_PAR_ERR	[47:2]	Address
SDP_RSP_NO_MTCH	[63:0]	Reserved
SPF_PRT_ERR	[63:0]	Reserved
SPF_ECC_ERR	[47:2]	Address
SDP_UNEXP_RETRY	[47:2]	Address
CNTR_OVFL	[63:0]	Reserved
CNTR_UNFL	[63:0]	Reserved
FTI_ND_ILL_REQ	[47:2]	Address
FTI_ND_ADDR_VIOL	[47:2]	Address
FTI_ND_SEC_VIOL	[47:2]	Address
HWA	[47:2]	Address
ST_PRT_ERR	[63:0]	Reserved
ST_ECC_ERR	[63:0]	Reserved
ST_TXN_ERR	[63:0]	Reserved

**MSR0000\_044F...MSRC000\_2163 [CS Machine Check Miscellaneous 0] (MCA::CS::MCA\_MISC0\_CS)**

Log miscellaneous information associated with errors.

\_instCS0\_n0\_aliasMSRLEGACY; MSR0000\_044F

\_instCS1\_n1\_aliasMSRLEGACY; MSR0000\_0453

\_instCS2\_n2\_aliasMSRLEGACY; MSR0000\_0457

\_instCS3\_n3\_aliasMSRLEGACY; MSR0000\_045B

\_instCS0\_n0\_aliasMSR; MSRC000\_2133

\_instCS1\_n1\_aliasMSR; MSRC000\_2143

\_instCS2\_n2\_aliasMSR; MSRC000\_2153

\_instCS3\_n3\_aliasMSR; MSRC000\_2163

Bits	Description
63	<b>Valid.</b> Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	<b>CntP.</b> Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	<b>Locked.</b> Reset: 0. 1=Writes to this register are ignored. This bit is set by BIOS to indicate that this register is not available for OS use. BIOS should set this bit if ThresholdIntType is set to SMI. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
60	<b>IntP.</b> Reset: 1. 1=ThresholdIntType can be used to generate interrupts. 0=ThresholdIntType and interrupt generation are not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::CS::MCA_MISC0_CS[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	<b>LvtOffset.</b> Reset: 0h. One per die. For error thresholding interrupts, specifies the address of the LVT entry in the APIC registers as follows: LVT address = (LvtOffset shifted left 4 bits) + 500h (see Core::X86::Apic::ExtendedInterruptLvtEntries). AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::CS::MCA_MISC0_CS[Locked]) ? Read-write : Read-only.
51	<b>CntEn.</b> Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::CS::MCA_MISC0_CS[Locked]) ? Read-write : Read-only.
50:49	<b>ThresholdIntType.</b> Reset: Cold,0h. Specifies the type of interrupt signaled when Ovrflw is set and IntP==1. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[ThresholdLvtOffset]). 10b = SMI trigger event. 11b = Reserved. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::CS::MCA_MISC0_CS[Locked]) ? Read-write : Read-only.
48	<b>Ovrflw.</b> Reset: Cold,0. Set by hardware when ErrCnt transitions from FFEh to FFFh. When this field is set, ErrCnt no longer increments. When this bit is set, the interrupt selected by the ThresholdIntType field is generated. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::CS::MCA_MISC0_CS[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	<b>ErrCnt.</b> Reset: Cold,000h. This is written by software to set the starting value of the error counter. This is incremented by hardware when errors are logged. When this counter overflows, it stays at FFFh (no rollover). The threshold value, written by software, is (FFFh - the desired error count (the number of errors necessary in order for an interrupt to be taken)); the desired error count of 0 (a write value of FFFh) is not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::CS::MCA_MISC0_CS[Locked]) ? Read-write : Read-only.
31:24	<b>BlkPtr.</b> Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

**MSRC000\_21[3...6]4 [CS Machine Check Configuration] (MCA::CS::MCA\_CONFIG\_CS)**

Reset: 0000_0000_0000_0125h.	
Controls configuration of the associated machine check bank.	
_instCS0_n0_aliasMSR; MSRC000_2134	
_instCS1_n1_aliasMSR; MSRC000_2144	
_instCS2_n2_aliasMSR; MSRC000_2154	
_instCS3_n3_aliasMSR; MSRC000_2164	
Bits	Description
63:41	Reserved.
40	<b>IntEn.</b> Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	<b>DeferredIntType.</b> Read-write. Reset: 0h. Specifies the type of interrupt signaled when a deferred error is logged. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[DeferredLvtOffset]). 10b = SMI trigger event. 11b = Reserved.
36:35	Reserved.
34	<b>LogDeferredInMcaStat.</b> Read-write. Reset: 0. Init: BIOS,1. 1=Log deferred errors in MCA::CS::MCA_STATUS_CS and MCA::CS::MCA_ADDR_CS in addition to MCA::CS::MCA_DESTAT_CS and MCA::CS::MCA_DEADDR_CS. 0=Only log deferred errors in MCA::CS::MCA_DESTAT_CS and MCA::CS::MCA_DEADDR_CS. This bit does not affect logging of deferred errors in MCA::CS::MCA_SYND_CS, MCA::CS::MCA_MISC0_CS.
33	Reserved.
32	<b>McaXEnable.</b> Read-write. Reset: 0. Init: BIOS,1. Check: 1. 1=Software has acknowledged support for the MCAX feature set. 0=Software has not acknowledged support for the MCAX feature set. All uncorrected and fatal errors will cause an ErrorEvent packet to be generated. Deferred error interrupts are configured via Core::X86::Msr::McaIntrCfg.
31:11	Reserved.
10	<b>IntPresent.</b> Read-only, Volatile. Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::CS::MCA_CONFIG_CS[IntEn].
9	<b>McaFruTextInMca.</b> Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	<b>McaLsbInStatusSupported.</b> Read-only. Reset: 1. 1=MCA::CS::MCA_CONFIG_CS[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	<b>DeferredIntTypeSupported.</b> Read-only. Reset: 1. 1=MCA::CS::MCA_CONFIG_CS[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::CS::MCA_CONFIG_CS[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	<b>DeferredErrorLoggingSupported.</b> Read-only. Reset: 1. 1=Deferred errors are supported in this MCA bank, and MCA::CS::MCA_CONFIG_CS[LogDeferredInMcaStat] controls the logging behavior of these errors. MCA::CS::MCA_DESTAT_CS and MCA::CS::MCA_DEADDR_CS are supported in this MCA bank. 0=Deferred errors are not supported in this bank.
1	Reserved.
0	<b>McaX.</b> Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::CS::MCA_MISC0_CS[BlkPtr] indicates the presence of the additional MISC registers, but is not used to determine their MSR numbers. Deferred error interrupt type is specifiable by MCA bank. MCA::CS::MCA_STATUS_CS[TCC] is present.

**MSRC000\_21[3...6]5 [CS IP Identification] (MCA::CS::MCA\_IPID\_CS)**

Reset: 0002_002E_0000_0000h.	
The MCA::CS::MCA_IPID_CS register is used by software to determine what IP type and revision is associated with the MCA bank.	
_instCS0_n0_aliasMSR; MSRC000_2135	
_instCS1_n1_aliasMSR; MSRC000_2145	
_instCS2_n2_aliasMSR; MSRC000_2155	
_instCS3_n3_aliasMSR; MSRC000_2165	
Bits	Description
63:48	<b>McaType</b> . Read-only. Reset: 0002h. The McaType of the MCA bank within this IP.
47:44	<b>InstanceIdHi</b> . Read-write. Reset: 0h. The HI value instance ID of this IP. This is initialized to a unique ID per instance of this register.
43:32	<b>HardwareID</b> . Read-only. Reset: 02Eh. The Hardware ID of the IP associated with this MCA bank.
31:0	<b>InstanceId</b> . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _instCS0_n0_aliasMSR: PSP,1F00_0000h
	Init: _instCS1_n1_aliasMSR: PSP,1F00_0100h
	Init: _instCS2_n2_aliasMSR: PSP,1F00_0200h
	Init: _instCS3_n3_aliasMSR: PSP,1F00_0300h

**MSRC000\_21[3...6]6 [CS Machine Check Syndrome] (MCA::CS::MCA\_SYND\_CS)**

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.	
Logs physical location information associated with error in MCA::CS::MCA_STATUS_CS Thread 0	
_instCS0_n0_aliasMSR; MSRC000_2136	
_instCS1_n1_aliasMSR; MSRC000_2146	
_instCS2_n2_aliasMSR; MSRC000_2156	
_instCS3_n3_aliasMSR; MSRC000_2166	
Bits	Description
63:32	<b>Syndrom</b> . Read-write, Volatile. Reset: Cold, 0000_0000h. Contains the syndrome, if any, associated with the error logged in MCA::CS::MCA_STATUS_CS. The low-order bit of the syndrome is stored in bit 0, and the syndrome has a length specified by MCA::CS::MCA_SYND_CS[Length]. The Syndrome field is only valid when MCA::CS::MCA_SYND_CS[Length] is not 0.
31:27	Reserved.
26:24	<b>ErrorPriority</b> . Read-write, Volatile. Reset: Cold, 0h. Encodes the priority of the error logged in MCA::CS::MCA_SYND_CS. 3'b000 = No error; 3'b001 = Reserved; 3'b010 = Corrected Error; 3'b011 = Deferred Error; 3'b100 = Uncorrected Error; 3'b101 = Fatal Error; all others reserved.
23:18	<b>Length</b> . Read-write, Volatile. Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::CS::MCA_SYND_CS[Syndrome]. A value of 0 indicates that there is no valid syndrome in MCA::CS::MCA_SYND_CS. For example, a syndrome length of 9 means that MCA::CS::MCA_SYND_CS[Syndrome] bits [8:0] contains a valid syndrome.
17:0	<b>ErrorInformation</b> . Read-write, Volatile. Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 60 [MCA_SYND_CS].

Table 60: MCA\_SYND\_CS

Error Type	Bits	Description
FTI_ILL_REQ	[17:0]	Reserved
FTI_ADDR_VIOL	[17:0]	Reserved
FTI_SEC_VIOL	[17:0]	Reserved
FTI_ILL_RSP	[17:0]	Reserved
FTI_RSP_NO_MTCH	[17:0]	Reserved
FTI_PAR_ERR	[6:0]	Reserved
SDP_PAR_ERR	[6:0]	Reserved

ATM_PAR_ERR	[6:0]	Reserved
SDP_RSP_NO_MTCH	[7:0]	Reserved
SPF_PRT_ERR	[17:0]	Reserved
SPF_ECC_ERR	[17:0]	Reserved
SDP_UNEXP_RETRY	[6:0]	Reserved
CNTR_OVFL	[17:0]	Reserved
CNTR_UNFL	[17:0]	Reserved
FTI_ND_ILL_REQ	[17:0]	Reserved
FTI_ND_ADDR_VIOL	[17:0]	Reserved
FTI_ND_SEC_VIOL	[17:0]	Reserved
HWA	[17:0]	Reserved
ST_PRT_ERR	[17:0]	Reserved
ST_ECC_ERR	[17:0]	Reserved
ST_TXN_ERR	[17:0]	Reserved

#### MSRC000\_21[3...6]8 [CS Machine Check Deferred Error Status] (MCA::CS::MCA\_DESTAT\_CS)

Reset: Cold,0000\_0000\_0000\_0000h.

Holds status information for the first deferred error seen in this bank.

\_instCS0\_n0\_aliasMSR; MSRC000\_2138

\_instCS1\_n1\_aliasMSR; MSRC000\_2148

\_instCS2\_n2\_aliasMSR; MSRC000\_2158

\_instCS3\_n3\_aliasMSR; MSRC000\_2168

Bits	Description
63	<b>Val.</b> Read-write, Volatile. Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	<b>Overflow.</b> Read-write, Volatile. Reset: Cold,0. 1=An error was detected while the valid bit (Val) was set; at least one error was not logged. Overflow is set independently of whether the existing error is overwritten. (See the section on error overwrite priorities.)
61:59	<b>RESERV4.</b> Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	<b>AddrV.</b> Read-write, Volatile. Reset: Cold,0. 1=MCA::CS::MCA_DEADDR_CS contains address information associated with the error.
57:54	<b>RESERV3.</b> Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	<b>SyndV.</b> Read-write, Volatile. Reset: Cold,0. 1=This error logged information in MCA::CS::MCA_SYND_CS. If MCA::CS::MCA_SYND_CS[ErrorPriority] is the same as the priority of the error in MCA::CS::MCA_STATUS_CS, then the information in MCA::CS::MCA_SYND_CS is associated with the error in MCA::CS::MCA_DESTAT_CS.
52:45	<b>RESERV2.</b> Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	<b>Deferred.</b> Read-write, Volatile. Reset: Cold,0. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; poison is created and an exception is deferred until the poison data is consumed.
43:30	<b>RESERV1.</b> Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	<b>AddrLsb.</b> Read-write, Volatile. Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::CS::MCA_ADDR_CS[ErrorAddr]. A value of 0 indicates that MCA::CS::MCA_ADDR_CS[63:0] contains a valid byte address. A value of 6 indicates that MCA::CS::MCA_ADDR_CS[63:6] contains a valid cache line address and that MCA::CS::MCA_ADDR_CS[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::CS::MCA_ADDR_CS[63:12] contain a valid 4KB memory page and that MCA::CS::MCA_ADDR_CS[11:0] should be ignored by error handling software.
23:22	<b>RESERV0.</b> Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	<b>ErrorCodeExt.</b> Read-write, Volatile. Reset: Cold,00h. Logs an extended error code when an error is detected. This model-specific field is used in conjunction with ErrorCode to identify the error sub-type for root cause analysis.
15:0	<b>ErrorCode.</b> Read-write, Volatile. Reset: Cold,0000h. Error code for this error.

**MSRC000\_21[3...6]9 [CS Deferred Error Address] (MCA::CS::MCA\_DEADDR\_CS)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

The MCA::CS::MCA\_DEADDR\_CS register stores the address associated with the error in MCA::CS::MCA\_DESTAT\_CS. The register is only meaningful if MCA::CS::MCA\_DESTAT\_CS[Val]=1 and MCA::CS::MCA\_DESTAT\_CS[AddrV]=1. The lowest valid bit of the address is defined by MCA::CS::MCA\_DESTAT\_CS[AddrLsb].

\_instCS0\_n0\_aliasMSR; MSRC000\_2139

\_instCS1\_n1\_aliasMSR; MSRC000\_2149

\_instCS2\_n2\_aliasMSR; MSRC000\_2159

\_instCS3\_n3\_aliasMSR; MSRC000\_2169

Bits	Description
63:0	<b>ErrorAddr.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::CS::MCA_DESTAT_CS. The lowest-order valid bit of the address is specified in MCA::CS::MCA_DESTAT_CS[AddrLsb].

**MSRC001\_041[3...6] [CS Machine Check Control Mask] (MCA::CS::MCA\_CTL\_MASK\_CS)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

Inhibit detection of an error source.

\_instCS0\_n0\_aliasMSR; MSRC001\_0413\_instCS1\_n1\_aliasMSR; MSRC001\_0414\_instCS2\_n2\_aliasMSR; MSRC001\_0415\_instCS3\_n3\_aliasMSR; MSRC001\_0416

Bits	Description
63:21	Reserved.
20	<b>ST_TXN_ERR.</b> Read-write. Reset: 0. Shadow Tag Transaction Error: An illegal Shadow Tag access occurred.
19	<b>ST_ECC_ERR.</b> Read-write. Reset: 0. Shadow Tag ECC Error: An ECC error occurred on a shadow tag array access.
18	<b>ST_PRT_ERR.</b> Read-write. Reset: 0. Shadow Tag Array Protocol Error: Indicates a Cache Coherence Issue.
17	<b>HWA.</b> Read-write. Reset: 0. Hardware Assert Error.
16	<b>FTI_ND_SEC_VIOL.</b> Read-write. Reset: 0. Security Violation: A security violation was detected on an incoming request from the transport layer on the request no data channel.
15	<b>FTI_ND_ADDR_VIOL.</b> Read-write. Reset: 0. Address Violation: An address violation was detected on an incoming request from the transport layer on the request no data channel.
14	<b>FTI_ND_ILL_REQ.</b> Read-write. Reset: 0. Illegal Request: An illegal request was received from the transport layer on the request no data channel.
13	<b>CNTR_UNFL.</b> Read-write. Reset: 0. Counter underflow error.
12	<b>CNTR_OVFL.</b> Read-write. Reset: 0. Counter overflow error.
11	<b>SDP_UNEXP_RETRY.</b> Read-write. Reset: 0. SDP read response had an unexpected RETRY error.
10	<b>SPF_ECC_ERR.</b> Read-write. Reset: 0. Probe Filter ECC Error: An ECC error occurred on a probe filter access.
9	<b>SPF_PRT_ERR.</b> Read-write. Reset: 0. Probe Filter Protocol Error: Indicates a Cache Coherence Issue.
8	<b>SDP_RSP_NO_MTCH.</b> Read-write. Reset: 0. SDP read response had no match in the CS queue.
7	<b>ATM_PAR_ERR.</b> Read-write. Reset: 0. Atomic Request Parity Error: Parity error on read of an atomic transaction.
6	<b>SDP_PAR_ERR.</b> Read-write. Reset: 0. Read Response Parity Error: Parity error on incoming read response data.
5	<b>FTI_PAR_ERR.</b> Read-write. Reset: 0. Request or Probe Parity Error: Parity error on incoming request or probe response data.
4	<b>FTI_RSP_NO_MTCH.</b> Read-write. Reset: 0. Unexpected Response: A response was received from the transport layer which does not match any request.
3	<b>FTI_ILL_RSP.</b> Read-write. Reset: 0. Illegal Response: An illegal response was received from the transport layer.
2	<b>FTI_SEC_VIOL.</b> Read-write. Reset: 0. Security Violation: A security violation was detected on an incoming request from the transport layer on the primary request channel.
1	<b>FTI_ADDR_VIOL.</b> Read-write. Reset: 0. Address Violation: An address violation was detected on an incoming request from the transport layer on the primary request channel.
0	<b>FTI_ILL_REQ.</b> Read-write. Reset: 0. Illegal Request: An illegal request was received from the transport layer on the primary request channel.



**MSRC000\_21[3...6]E [CS Machine Check Syndrome Extended] (MCA::CS::MCA\_SYND1\_CS)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::CS::MCA\_STATUS\_CS Thread 0

\_instCS0\_n0\_aliasMSR; MSRC000\_213E\_instCS1\_n1\_aliasMSR; MSRC000\_214E\_instCS2\_n2\_aliasMSR; MSRC000\_215E\_instCS3\_n3\_aliasMSR; MSRC000\_216E

Bits	Description
63:0	<b>Syndrone.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::CS::MCA_SYND1_CS register stores information associated with the error in MCA::CS::MCA_STATUS_CS or MCA_DESTAT. The register is meaningful if MCA::CS::MCA_STATUS_CS[SyndV]=1. When MCA::CS::MCA_CONFIG_CS[McaFruTextInMca]=1, MCA::CS::MCA_SYND1_CS stores ASCII FruText associated with the error.

**MSRC000\_21[3...6]F [CS Machine Check Syndrome Extended] (MCA::CS::MCA\_SYND2\_CS)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::CS::MCA\_STATUS\_CS Thread 0

\_instCS0\_n0\_aliasMSR; MSRC000\_213F\_instCS1\_n1\_aliasMSR; MSRC000\_214F\_instCS2\_n2\_aliasMSR; MSRC000\_215F\_instCS3\_n3\_aliasMSR; MSRC000\_216F

Bits	Description
63:0	<b>Syndrone.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::CS::MCA_SYND2_CS register stores information associated with the error in MCA::CS::MCA_STATUS_CS or MCA_DESTAT. The register is meaningful if MCA::CS::MCA_STATUS_CS[SyndV]=1. When MCA::CS::MCA_CONFIG_CS[McaFruTextInMca]=1, MCA::CS::MCA_SYND2_CS stores ASCII FruText associated with the error.

**3.2.5.9 PIE****MSR0000\_046C...MSRC000\_21B0 [PIE Machine Check Control] (MCA::PIE::MCA\_CTL\_PIE)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

0=Disables error reporting for the corresponding error. 1=Enables error reporting via machine check exception for the corresponding error. The MCA::PIE::MCA\_CTL\_PIE register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG\_CTL. Does not affect error detection, correction, or logging.

\_instPIE0\_n0\_aliasMSRLEGACY; MSR0000\_046C\_instPIE0\_n0\_aliasMSR; MSRC000\_21B0

Bits	Description
63:9	Reserved.
8	<b>DSM_ACT.</b> Read-write. Reset: 0.
7	<b>RSLVFCI.</b> Read-write. Reset: 0. Register access during DF Cstate:
6	<b>CNLI.</b> Read-write. Reset: 0. An SRAM ECC error in the CNLI block.
5	<b>WDT.</b> Read-write. Reset: 0. Watch Dog Timer: A watch dog timer expired.
4	<b>DEF.</b> Read-write. Reset: 0. A deferred error was detected in the DF.
3	<b>FTI_DAT_STAT.</b> Read-write. Reset: 0. Poison data consumption: Poison data was written to an internal PIE register.
2	<b>GMI.</b> Read-write. Reset: 0. Reserved .
1	<b>CSW.</b> Read-write. Reset: 0. Register security violation: A security violation was detected on an access to an internal PIE register.
0	<b>HW_ASSERT.</b> Read-write. Reset: 0. Hardware Assert: A hardware assert was detected.



**MSR0000\_046D...MSRC000\_21B1 [PIE Machine Check Status] (MCA::PIE::MCA\_STATUS\_PIE)**

Reset: Cold,0000\_0000\_0000\_0000h.

Logs information associated with errors.

\_instPIE0\_n0\_aliasMSRLEGACY; MSR0000\_046D

\_instPIE0\_n0\_aliasMSR; MSRC000\_21B1

Bits	Description
63	<b>Val.</b> Reset: Cold,0. 1=A valid error has been detected. This bit should be cleared by software after the register has been read. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
62	<b>Overflow.</b> Reset: Cold,0. 1=An error was detected while the valid bit (Val) was set; at least one error was not logged. Overflow is set independently of whether the existing error is overwritten. See 3.1.3 [Machine Check Errors]. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
61	<b>UC.</b> Reset: Cold,0. 1=The error was not corrected by hardware. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
60	<b>En.</b> Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::PIE::MCA_CTL_PIE. This bit is a copy of bit in MCA::PIE::MCA_CTL_PIE for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	<b>MiscV.</b> Reset: Cold,0. 1=Valid thresholding in MCA::PIE::MCA_MISC0_PIE. In certain modes, MISC registers are owned by platform firmware and will RAZ when read by non-SMM code. Therefore, it is possible for MiscV=1 and the MISC register to read as all zeros. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
58	<b>AddrV.</b> Reset: Cold,0. 1=MCA::PIE::MCA_ADDR_PIE contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	<b>PCC.</b> Reset: Cold,0. 1=Hardware context held by the processor may have been corrupted. Continued operation of the system may have unpredictable results. The error is not recoverable or survivable, and the system should be reinitialized. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
56	<b>ErrCoreIdVal.</b> Reset: Cold,0. 1=The ErrCoreId field is valid. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
55	<b>TCC.</b> Reset: Cold,0. 1=Hardware context of the process thread to which the error was reported may have been corrupted. Continued operation of the thread may have unpredictable results. The thread must be terminated. Only meaningful when MCA::PIE::MCA_STATUS_PIE[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	<b>RESERV54.</b> Reset: Cold,0. MCA_STATUS Register Reserved bit. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
53	<b>SyndV.</b> Reset: Cold,0. 1=This error logged information in MCA::PIE::MCA_SYND_PIE. If MCA::PIE::MCA_SYND_PIE[ErrorPriority] is the same as the priority of the error in MCA::PIE::MCA_STATUS_PIE, then the information in MCA::PIE::MCA_SYND_PIE is associated with the error in MCA::PIE::MCA_STATUS_PIE. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	<b>RESERV47.</b> Reset: Cold,00h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
46	<b>CECC.</b> Reset: Cold,0. 1=The error was a correctable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
45	<b>UECC.</b> Reset: Cold,0. 1=The error was an uncorrectable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.

44	<b>Deferred.</b> Reset: Cold,0. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; an exception is deferred until the erroneous data is consumed. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
43	<b>Poison.</b> Reset: Cold,0. 1=The error was the result of attempting to consume poisoned data. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
42:41	<b>RESERV41.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
40	<b>Scrub.</b> Reset: Cold,0. 1=The error was the result of a scrub operation. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
39:38	<b>RESERV38.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
37:32	<b>ErrCoreId.</b> Reset: Cold,00h. When ErrCoreIdVal=1 this field indicates which core within the processor is associated with the error; Otherwise this field is reserved. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
31:30	<b>RESERV30.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
29:24	<b>AddrLsb.</b> Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::PIE::MCA_ADDR_PIE[ErrorAddr]. A value of 0 indicates that MCA::PIE::MCA_ADDR_PIE[63:0] contains a valid byte address. A value of 6 indicates that MCA::PIE::MCA_ADDR_PIE[63:6] contains a valid cache line address and that MCA::PIE::MCA_ADDR_PIE[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::PIE::MCA_ADDR_PIE[63:12] contain a valid 4KB memory page and that MCA::PIE::MCA_ADDR_PIE[11:0] should be ignored by error handling software. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
23:22	<b>RESERV22.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
21:16	<b>ErrorCodeExt.</b> Reset: Cold,00h. Extended Error Code. This field is used to identify the error type for root cause analysis. This field indicates which bit position in MCA::PIE::MCA_CTL_PIE enables error reporting for the logged error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
15:0	<b>ErrorCode.</b> Reset: Cold,0000h. Error code for this error. See 3.1.3.3 [Error Codes] for details on decoding this field. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.

Table 61: MCA\_STATUS\_PIE

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
HW_ASSER T	0x0	1	1	1	0	0	0
CSW	0x1	0	0	0	1	0	0
GMI	0x2	0/1	0/1	0/1	0	0	0
FTI_DAT_S TAT	0x3	1	1	1	0	0	0
DEF	0x4	0	0	0	1	0	0
WDT	0x5	1	1	1	0	0	1
CNLI	0x6	0/1	0/1	0/1	0	0	0
RSLVFCI	0x7	0	0	0	0/1	0	0
DSM_ACT	0x8	1	1	1	0	0	0

**MSR0000\_046E...MSRC000\_21B2 [PIE Machine Check Address] (MCA::PIE::MCA\_ADDR\_PIE)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

MCA::PIE::MCA\_ADDR\_PIE stores an address and other information associated with the error in MCA::PIE::MCA\_STATUS\_PIE. The register is only meaningful if MCA::PIE::MCA\_STATUS\_PIE[Val]=1 and MCA::PIE::MCA\_STATUS\_PIE[AddrV]=1.

\_instPIE0\_n0\_aliasMSRLEGACY; MSR0000\_046E

\_instPIE0\_n0\_aliasMSR; MSRC000\_21B2

Bits	Description
63:0	<b>ErrorAddr.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. Unless otherwise specified by an error, contains the address associated with the error logged in MCA::PIE::MCA_STATUS_PIE. For physical addresses, the most significant bit is given by Core::X86::Cpuid::LongModeInfo[PhysAddrSize].

Table 62: MCA\_ADDR\_PIE

Error Type	Bits	Description
HW_ASSERT	[63:0]	Reserved
CSW	[63:0]	Reserved
GMI	[63:0]	Reserved
FTI_DAT_STAT	[63:0]	Reserved
DEF	[63:0]	Reserved
WDT	[63:0]	Reserved
CNLI	[63:0]	Reserved
RSLVFCI	[63:0]	Reserved
DSM_ACT	[63:0]	Reserved

**MSR0000\_046F...MSRC000\_21B3 [PIE Machine Check Miscellaneous 0] (MCA::PIE::MCA\_MISC0\_PIE)**

Log miscellaneous information associated with errors.

\_instPIE0\_n0\_aliasMSRLEGACY; MSR0000\_046F

\_instPIE0\_n0\_aliasMSR; MSRC000\_21B3

Bits	Description
63	<b>Valid.</b> Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	<b>CntP.</b> Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	<b>Locked.</b> Reset: 0. 1=Writes to this register are ignored. This bit is set by BIOS to indicate that this register is not available for OS use. BIOS should set this bit if ThresholdIntType is set to SMI. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
60	<b>IntP.</b> Reset: 1. 1=ThresholdIntType can be used to generate interrupts. 0=ThresholdIntType and interrupt generation are not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::PIE::MCA_MISC0_PIE[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	<b>LvtOffset.</b> Reset: 0h. One per die. For error thresholding interrupts, specifies the address of the LVT entry in the APIC registers as follows: LVT address = (LvtOffset shifted left 4 bits) + 500h (see Core::X86::Apic::ExtendedInterruptLvtEntries). AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::PIE::MCA_MISC0_PIE[Locked]) ? Read-write : Read-only.
51	<b>CntEn.</b> Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::PIE::MCA_MISC0_PIE[Locked]) ? Read-write : Read-only.
50:49	<b>ThresholdIntType.</b> Reset: Cold,0h. Specifies the type of interrupt signaled when Ovrflw is set and IntP==1. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[ThresholdLvtOffset]). 10b = SMI trigger event. 11b = Reserved. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::PIE::MCA_MISC0_PIE[Locked]) ? Read-write : Read-only.
48	<b>Ovrflw.</b> Reset: Cold,0. Set by hardware when ErrCnt transitions from FFEh to FFFh. When this field is set, ErrCnt no longer increments. When this bit is set, the interrupt selected by the ThresholdIntType field is generated. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::PIE::MCA_MISC0_PIE[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	<b>ErrCnt.</b> Reset: Cold,000h. This is written by software to set the starting value of the error counter. This is incremented by hardware when errors are logged. When this counter overflows, it stays at FFFh (no rollover). The threshold value, written by software, is (FFFh - the desired error count (the number of errors necessary in order for an interrupt to be taken)); the desired error count of 0 (a write value of FFFh) is not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::PIE::MCA_MISC0_PIE[Locked]) ? Read-write : Read-only.
31:24	<b>BlkPtr.</b> Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

**MSRC000\_21B4 [PIE Machine Check Configuration] (MCA::PIE::MCA\_CONFIG\_PIE)**

Reset: 0000_0002_0000_0125h.	
Controls configuration of the associated machine check bank.	
_instPIE0_n0_aliasMSR; MSRC000_21B4	
Bits	Description
63:41	Reserved.
40	<b>IntEn.</b> Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	<b>DeferredIntType.</b> Read-write. Reset: 0h. Specifies the type of interrupt signaled when a deferred error is logged. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[DeferredLvtOffset]). 10b = SMI trigger event. 11b = Reserved.
36:35	Reserved.
34	<b>LogDeferredInMcaStat.</b> Read-write. Reset: 0. Init: BIOS,1. 1=Log deferred errors in MCA::PIE::MCA_STATUS_PIE and MCA::PIE::MCA_ADDR_PIE in addition to MCA::PIE::MCA_DESTAT_PIE and MCA::PIE::MCA_DEADDR_PIE. 0=Only log deferred errors in MCA::PIE::MCA_DESTAT_PIE and MCA::PIE::MCA_DEADDR_PIE. This bit does not affect logging of deferred errors in MCA::PIE::MCA_SYND_PIE, MCA::PIE::MCA_MISC0_PIE.
33	Reserved.
32	<b>McaXEnable.</b> Read-write. Reset: 0. Init: BIOS,1. Check: 1. 1=Software has acknowledged support for the MCAX feature set. 0=Software has not acknowledged support for the MCAX feature set. All uncorrected and fatal errors will cause an ErrorEvent packet to be generated. Deferred error interrupts are configured via Core::X86::Msr::McaIntrCfg.
31:11	Reserved.
10	<b>IntPresent.</b> Read-only, Volatile. Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::PIE::MCA_CONFIG_PIE[IntEn].
9	<b>McaFruTextInMca.</b> Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	<b>McaLsbInStatusSupported.</b> Read-only. Reset: 1. 1=MCA::PIE::MCA_CONFIG_PIE[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	<b>DeferredIntTypeSupported.</b> Read-only. Reset: 1. 1=MCA::PIE::MCA_CONFIG_PIE[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::PIE::MCA_CONFIG_PIE[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	<b>DeferredErrorLoggingSupported.</b> Read-only. Reset: 1. 1=Deferred errors are supported in this MCA bank, and MCA::PIE::MCA_CONFIG_PIE[LogDeferredInMcaStat] controls the logging behavior of these errors. MCA::PIE::MCA_DESTAT_PIE and MCA::PIE::MCA_DEADDR_PIE are supported in this MCA bank. 0=Deferred errors are not supported in this bank.
1	Reserved.
0	<b>McaX.</b> Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::PIE::MCA_MISC0_PIE[BlkPtr] indicates the presence of the additional MISC registers, but is not used to determine their MSR numbers. Deferred error interrupt type is specifiable by MCA bank. MCA::PIE::MCA_STATUS_PIE[TCC] is present.

**MSRC000\_21B5 [PIE IP Identification] (MCA::PIE::MCA\_IPID\_PIE)**

Reset: 0001\_002E\_0000\_0000h.

The MCA::PIE::MCA\_IPID\_PIE register is used by software to determine what IP type and revision is associated with the MCA bank.

\_instPIE0\_n0\_aliasMSR; MSRC000\_21B5

Bits	Description
63:48	<b>McaType</b> . Read-only. Reset: 0001h. The McaType of the MCA bank within this IP.
47:44	<b>InstanceIdHi</b> . Read-write. Reset: 0h. The HI value instance ID of this IP. This is initialized to a unique ID per instance of this register.
43:32	<b>HardwareID</b> . Read-only. Reset: 02Eh. The Hardware ID of the IP associated with this MCA bank.
31:0	<b>InstanceId</b> . Read-write. Reset: 0000_0000h. Init: PSP,1F00_1100h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.

**MSRC000\_21B6 [PIE Machine Check Syndrome] (MCA::PIE::MCA\_SYND\_PIE)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::PIE::MCA\_STATUS\_PIE Thread 0

\_instPIE0\_n0\_aliasMSR; MSRC000\_21B6

Bits	Description
63:32	<b>Syndrom</b> . Read-write, Volatile. Reset: Cold, 0000_0000h. Contains the syndrome, if any, associated with the error logged in MCA::PIE::MCA_STATUS_PIE. The low-order bit of the syndrome is stored in bit 0, and the syndrome has a length specified by MCA::PIE::MCA_SYND_PIE[Length]. The Syndrome field is only valid when MCA::PIE::MCA_SYND_PIE[Length] is not 0.
31:27	Reserved.
26:24	<b>ErrorPriority</b> . Read-write, Volatile. Reset: Cold, 0h. Encodes the priority of the error logged in MCA::PIE::MCA_SYND_PIE. 3'b000 = No error; 3'b001 = Reserved; 3'b010 = Corrected Error; 3'b011 = Deferred Error; 3'b100 = Uncorrected Error; 3'b101 = Fatal Error; all others reserved.
23:18	<b>Length</b> . Read-write, Volatile. Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::PIE::MCA_SYND_PIE[Syndrome]. A value of 0 indicates that there is no valid syndrome in MCA::PIE::MCA_SYND_PIE. For example, a syndrome length of 9 means that MCA::PIE::MCA_SYND_PIE[Syndrome] bits [8:0] contains a valid syndrome.
17:0	<b>ErrorInformation</b> . Read-write, Volatile. Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 63 [MCA_SYND_PIE].

Table 63: MCA\_SYND\_PIE

Error Type	Bits	Description
HW_ASSERT	[17:16]	Reserved
	[15:8]	Block Instance ID
	[7:6]	Reserved
	[5:0]	Hardware Assert ID
CSW	[17]	Reserved
	[16:13]	Reserved
	[12:0]	Reserved
GMI	[17:0]	Reserved
FTI_DAT_STAT	[3:0]	Reserved
DEF	[17:0]	Reserved
WDT	[17:5]	Reserved
	[3:1]	Reserved
	[0:0]	Reserved
CNLI	[17:8]	Reserved
	[7:6]	Reserved



	[5:4]	Reserved
	[3:3]	Reserved
	[2:0]	Reserved
RSLVFCI	[17]	Reserved
	[16]	Reserved
	[15]	Reserved
	[14]	Reserved
	[13]	Reserved
	[12:0]	Reserved
DSM_ACT	[17:0]	Reserved

**MSRC000\_21B8 [PIE Machine Check Deferred Error Status] (MCA::PIE::MCA\_DESTAT\_PIE)**

Reset: Cold,0000\_0000\_0000\_0000h.

Holds status information for the first deferred error seen in this bank.

\_instPIE0\_n0\_aliasMSR; MSRC000\_21B8

Bits	Description
63	<b>Val.</b> Read-write, Volatile. Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	<b>Overflow.</b> Read-write, Volatile. Reset: Cold,0. 1=An error was detected while the valid bit (Val) was set; at least one error was not logged. Overflow is set independently of whether the existing error is overwritten. (See the section on error overwrite priorities.)
61:59	<b>RESERV4.</b> Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	<b>AddrV.</b> Read-write, Volatile. Reset: Cold,0. 1=MCA::PIE::MCA_DEADDR_PIE contains address information associated with the error.
57:54	<b>RESERV3.</b> Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	<b>SyndV.</b> Read-write, Volatile. Reset: Cold,0. 1=This error logged information in MCA::PIE::MCA_SYND_PIE. If MCA::PIE::MCA_SYND_PIE[ErrorPriority] is the same as the priority of the error in MCA::PIE::MCA_STATUS_PIE, then the information in MCA::PIE::MCA_SYND_PIE is associated with the error in MCA::PIE::MCA_DESTAT_PIE.
52:45	<b>RESERV2.</b> Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	<b>Deferred.</b> Read-write, Volatile. Reset: Cold,0. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; poison is created and an exception is deferred until the poison data is consumed.
43:30	<b>RESERV1.</b> Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	<b>AddrLsb.</b> Read-write, Volatile. Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::PIE::MCA_ADDR_PIE[ErrorAddr]. A value of 0 indicates that MCA::PIE::MCA_ADDR_PIE[63:0] contains a valid byte address. A value of 6 indicates that MCA::PIE::MCA_ADDR_PIE[63:6] contains a valid cache line address and that MCA::PIE::MCA_ADDR_PIE[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::PIE::MCA_ADDR_PIE[63:12] contain a valid 4KB memory page and that MCA::PIE::MCA_ADDR_PIE[11:0] should be ignored by error handling software.
23:22	<b>RESERV0.</b> Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	<b>ErrorCodeExt.</b> Read-write, Volatile. Reset: Cold,00h. Logs an extended error code when an error is detected. This model-specific field is used in conjunction with ErrorCode to identify the error sub-type for root cause analysis.
15:0	<b>ErrorCode.</b> Read-write, Volatile. Reset: Cold,0000h. Error code for this error.

**MSRC000\_21B9 [PIE Deferred Error Address] (MCA::PIE::MCA\_DEADDR\_PIE)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

The MCA::PIE::MCA\_DEADDR\_PIE register stores the address associated with the error in MCA::PIE::MCA\_DESTAT\_PIE. The register is only meaningful if MCA::PIE::MCA\_DESTAT\_PIE[Val]=1 and MCA::PIE::MCA\_DESTAT\_PIE[AddrV]=1. The lowest valid bit of the address is defined by MCA::PIE::MCA\_DESTAT\_PIE[AddrLsb].

\_instPIE0\_n0\_aliasMSR; MSRC000\_21B9

Bits	Description
63:0	<b>ErrorAddr.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::PIE::MCA_DESTAT_PIE. The lowest-order valid bit of the address is specified in MCA::PIE::MCA_DESTAT_PIE[AddrLsb].

**MSRC001\_041B [PIE Machine Check Control Mask] (MCA::PIE::MCA\_CTL\_MASK\_PIE)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

Inhibit detection of an error source.

\_instPIE0\_n0\_aliasMSR; MSRC001\_041B

Bits	Description
63:9	Reserved.
8	<b>DSM_ACT.</b> Read-write. Reset: 0.
7	<b>RSLVFCI.</b> Read-write. Reset: 0. Register access during DF Cstate:
6	<b>CNLI.</b> Read-write. Reset: 0. An SRAM ECC error in the CNLI block.
5	<b>WDT.</b> Read-write. Reset: 0. Watch Dog Timer: A watch dog timer expired.
4	<b>DEF.</b> Read-write. Reset: 0. A deferred error was detected in the DF.
3	<b>FTI_DAT_STAT.</b> Read-write. Reset: 0. Poison data consumption: Poison data was written to an internal PIE register.
2	<b>GMI.</b> Read-write. Reset: 0. Reserved .
1	<b>CSW.</b> Read-write. Reset: 0. Register security violation: A security violation was detected on an access to an internal PIE register.
0	<b>HW_ASSERT.</b> Read-write. Reset: 0. Hardware Assert: A hardware assert was detected.

**MSRC000\_21BE [PIE Machine Check Syndrome Extended] (MCA::PIE::MCA\_SYND1\_PIE)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::PIE::MCA\_STATUS\_PIE Thread 0

\_instPIE0\_n0\_aliasMSR; MSRC000\_21BE

Bits	Description
63:0	<b>Syndrome.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::PIE::MCA_SYND1_PIE register stores information associated with the error in MCA::PIE::MCA_STATUS_PIE or MCA_DESTAT. The register is meaningful if MCA::PIE::MCA_STATUS_PIE[SyndV]=1. When MCA::PIE::MCA_CONFIG_PIE[McaFruTextInMca]=1, MCA::PIE::MCA_SYND1_PIE stores ASCII FruText associated with the error.

**MSRC000\_21BF [PIE Machine Check Syndrome Extended] (MCA::PIE::MCA\_SYND2\_PIE)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::PIE::MCA\_STATUS\_PIE Thread 0

\_instPIE0\_n0\_aliasMSR; MSRC000\_21BF

Bits	Description
63:0	<b>Syndrome.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::PIE::MCA_SYND2_PIE register stores information associated with the error in MCA::PIE::MCA_STATUS_PIE or MCA_DESTAT. The register is meaningful if MCA::PIE::MCA_STATUS_PIE[SyndV]=1. When MCA::PIE::MCA_CONFIG_PIE[McaFruTextInMca]=1, MCA::PIE::MCA_SYND2_PIE stores ASCII FruText associated with the error.



## 3.2.5.10 UMC

MSR0000_043C...MSRC000_2120 [UMC Machine Check Control] (MCA::UMC::MCA_CTL_UMC)	
Read-write. Reset: 0000_0000_0000_0000h.	
0=Disables error reporting for the corresponding error. 1=Enables error reporting via machine check exception for the corresponding error. The MCA::UMC::MCA_CTL_UMC register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG_CTL. Does not affect error detection, correction, or logging.	
_instUMC_n0_umc0_aliasMSRLEGACY; MSR0000_043C	
_instUMC_n1_umc1_aliasMSRLEGACY; MSR0000_0440	
_instUMC_n2_umc0_aliasMSRLEGACY; MSR0000_0444	
_instUMC_n3_umc1_aliasMSRLEGACY; MSR0000_0448	
_instUMC_n0_umc0_aliasMSR; MSRC000_20F0	
_instUMC_n1_umc1_aliasMSR; MSRC000_2100	
_instUMC_n2_umc0_aliasMSR; MSRC000_2110	
_instUMC_n3_umc1_aliasMSR; MSRC000_2120	
Bits	Description
63:13	Reserved.
12	<b>LinkEccErr.</b> Read-write. Reset: 0. Link ECC error. An ECC error occurred on a DRAM link
11	<b>RdCrcErr.</b> Read-write. Reset: 0. Read CRC error. CRC error occurred on a DRAM read from any subchannel
10	<b>ThrttlErr.</b> Read-write. Reset: 0. Indicates that UMC is throttling
9	<b>EcsErr.</b> Read-write. Reset: 0. ECS Error. Indicates that a device exceeded the ECS Error Threshold Count.
8	<b>EcsRowErr.</b> Read-write. Reset: 0. ECS Row Error. Indicates that a single device row exceeded four code word errors.
7	<b>AesSramEccErr.</b> Read-write. Reset: 0. AES SRAM ECC error. An ECC error occurred on a AES SRAM in the processor.
6	<b>DcqSramEccErr.</b> Read-write. Reset: 0. DCQ SRAM ECC error. An ECC error occurred on a DCQ SRAM in the processor.
5	<b>WriteDataCrcErr.</b> Read-write. Reset: 0. Write data CRC error. A write data CRC error occurred on the DRAM data bus.
4	<b>AddressCommandParityErr.</b> Read-write. Reset: 0. Address/Command parity error. A parity error occurred on the DRAM address/command bus.
3	<b>ApbErr.</b> Read-write. Reset: 0. Advanced peripheral bus error. An error occurred on the advanced peripheral bus.
2	<b>SdpParityErr.</b> Read-write. Reset: 0. SDP parity error. A parity error was detected on write data from the data fabric in the processor.
1	<b>WriteDataPoisonErr.</b> Read-write. Reset: 0. Data poison error. The system tried to write poison data to DRAM and either DRAM does not support ECC or UMC_CH.EccCtrl.WrEccEn is cleared.
0	<b>DramEccErr.</b> Read-write. Reset: 0. DRAM ECC error. An ECC error occurred on a DRAM read.

**MSR0000\_043D...MSRC000\_2121 [UMC Machine Check Status] (MCA::UMC::MCA\_STATUS\_UMC)**

Reset: Cold,0000\_0000\_0000\_0000h.

Logs information associated with errors.

\_instUMC\_n0\_umc0\_aliasMSRLEGACY; MSR0000\_043D

\_instUMC\_n1\_umc1\_aliasMSRLEGACY; MSR0000\_0441

\_instUMC\_n2\_umc0\_aliasMSRLEGACY; MSR0000\_0445

\_instUMC\_n3\_umc1\_aliasMSRLEGACY; MSR0000\_0449

\_instUMC\_n0\_umc0\_aliasMSR; MSRC000\_20F1

\_instUMC\_n1\_umc1\_aliasMSR; MSRC000\_2101

\_instUMC\_n2\_umc0\_aliasMSR; MSRC000\_2111

\_instUMC\_n3\_umc1\_aliasMSR; MSRC000\_2121

Bits	Description
63	<b>Val.</b> Reset: Cold,0. 1=A valid error has been detected. This bit should be cleared by software after the register has been read. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
62	<b>Overflow.</b> Reset: Cold,0. 1=An error was detected while the valid bit (Val) was set; at least one error was not logged. Overflow is set independently of whether the existing error is overwritten. See 3.1.3 [Machine Check Errors]. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
61	<b>UC.</b> Reset: Cold,0. 1=The error was not corrected by hardware. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
60	<b>En.</b> Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::UMC::MCA_CTL_UMC. This bit is a copy of bit in MCA::UMC::MCA_CTL_UMC for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	<b>MiscV.</b> Reset: Cold,0. 1=Valid thresholding in MCA::UMC::MCA_MISC0_UMC. In certain modes, MISC registers are owned by platform firmware and will RAZ when read by non-SMM code. Therefore, it is possible for MiscV=1 and the MISC register to read as all zeros. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
58	<b>AddrV.</b> Reset: Cold,0. 1=MCA::UMC::MCA_ADDR_UMC contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	<b>PCC.</b> Reset: Cold,0. 1=Hardware context held by the processor may have been corrupted. Continued operation of the system may have unpredictable results. The error is not recoverable or survivable, and the system should be reinitialized. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
56	<b>ErrCoreIdVal.</b> Reset: Cold,0. 1=The ErrCoreId field is valid. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
55	<b>TCC.</b> Reset: Cold,0. 1=Hardware context of the process thread to which the error was reported may have been corrupted. Continued operation of the thread may have unpredictable results. The thread must be terminated. Only meaningful when MCA::UMC::MCA_STATUS_UMC[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	<b>RESERV54.</b> Reset: Cold,0. MCA_STATUS Register Reserved bit. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
53	<b>SyndV.</b> Reset: Cold,0. 1=This error logged information in MCA::UMC::MCA_SYND_UMC. If MCA::UMC::MCA_SYND_UMC[ErrorPriority] is the same as the priority of the error in MCA::UMC::MCA_STATUS_UMC, then the information in MCA::UMC::MCA_SYND_UMC is associated with the error in MCA::UMC::MCA_STATUS_UMC. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	<b>RESERV47.</b> Reset: Cold,00h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.

46	<b>CECC.</b> Reset: Cold,0. 1=The error was a correctable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
45	<b>UECC.</b> Reset: Cold,0. 1=The error was an uncorrectable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
44	<b>Deferred.</b> Reset: Cold,0. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; an exception is deferred until the erroneous data is consumed. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
43	<b>Poison.</b> Reset: Cold,0. 1=The error was the result of attempting to consume poisoned data. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
42:41	<b>RESERV41.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
40	<b>Scrub.</b> Reset: Cold,0. 1=The error was the result of a scrub operation. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
39:38	<b>RESERV38.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
37:32	<b>ErrCoreId.</b> Reset: Cold,00h. When ErrCoreIdVal=1 this field indicates which core within the processor is associated with the error; Otherwise this field is reserved. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
31:30	<b>RESERV30.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
29:24	<b>AddrLsb.</b> Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::UMC::MCA_ADDR_UMC[ErrorAddr]. A value of 0 indicates that MCA::UMC::MCA_ADDR_UMC[63:0] contains a valid byte address. A value of 6 indicates that MCA::UMC::MCA_ADDR_UMC[63:6] contains a valid cache line address and that MCA::UMC::MCA_ADDR_UMC[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::UMC::MCA_ADDR_UMC[63:12] contain a valid 4KB memory page and that MCA::UMC::MCA_ADDR_UMC[11:0] should be ignored by error handling software. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
23:22	<b>RESERV22.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
21:16	<b>ErrorCodeExt.</b> Reset: Cold,00h. Extended Error Code. This field is used to identify the error type for root cause analysis. This field indicates which bit position in MCA::UMC::MCA_CTL_UMC enables error reporting for the logged error. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
15:0	<b>ErrorCode.</b> Reset: Cold,0000h. Error code for this error. See 3.1.3.3 [Error Codes] for details on decoding this field. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.

Table 64: MCA\_STATUS\_UMC

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
DramEccErr	0x0	0/1	0/1	0/1	0/1	0	1
WriteDataPoisonErr	0x1	1	1	1	0	0	0
SdpParityErr	0x2	1	1	1	0	0	0
ApbErr	0x3	1	1	1	0	0	1

AddressCommandParityError	0x4	0/1	0/1	0/1	0	0	0/1
WriteDataCrcError	0x5	0/1	0/1	0/1	0	0	0/1
DcqSramEccError	0x6	0/1	0/1	0/1	0	0	0
AesSramEccError	0x7	0/1	0/1	0/1	0	0	0
EcsRowError	0x8	0	0	0	0	0	0
EcsError	0x9	0	0	0	0	0	0
ThrttlError	0xa	0	0	0	0	0	0
RdCrcError	0xb	0	0	0	1	0	1
LinkEccError	0xc	0/1	0/1	0/1	0/1	0	1

**MSR0000\_043E...MSRC000\_2122 [UMC Machine Check Address] (MCA::UMC::MCA\_ADDR\_UMC)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

MCA::UMC::MCA\_ADDR\_UMC stores an address and other information associated with the error in MCA::UMC::MCA\_STATUS\_UMC. The register is only meaningful if MCA::UMC::MCA\_STATUS\_UMC[Val]=1 and MCA::UMC::MCA\_STATUS\_UMC[AddrV]=1.

\_instUMC\_n0\_umc0\_aliasMSRLEGACY; MSR0000\_043E

\_instUMC\_n1\_umc1\_aliasMSRLEGACY; MSR0000\_0442

\_instUMC\_n2\_umc0\_aliasMSRLEGACY; MSR0000\_0446

\_instUMC\_n3\_umc1\_aliasMSRLEGACY; MSR0000\_044A

\_instUMC\_n0\_umc0\_aliasMSR; MSRC000\_20F2

\_instUMC\_n1\_umc1\_aliasMSR; MSRC000\_2102

\_instUMC\_n2\_umc0\_aliasMSR; MSRC000\_2112

\_instUMC\_n3\_umc1\_aliasMSR; MSRC000\_2122

Bits	Description
63:0	<b>ErrorAddr.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. Unless otherwise specified by an error, contains the address associated with the error logged in MCA::UMC::MCA_STATUS_UMC. For physical addresses, the most significant bit is given by Core::X86::Cpuid::LongModeInfo[PhysAddrSize].

Table 65: MCA\_ADDR\_UMC

Error Type	Bits	Description
DramEccError	[55:39] [39:4]	Reserved Reserved
WriteDataPoisonError	[63:0]	Reserved
SdpParityError	[63:0]	Reserved
ApbError	[55:30] [29:0]	Reserved Reserved
AddressCommandParityError	[55:38] [37:36] [35:32] [31:0]	Reserved Reserved Chip Select Reserved
WriteDataCrcError	[55:38] [37:36] [35:32] [31:0]	Reserved Reserved Chip Select Reserved
DcqSramEccError	[63:0]	Reserved
AesSramEccError	[63:0]	Reserved

EcsRowErr	[63:0]	Reserved
EcsErr	[63:0]	Reserved
ThrttlErr	[63:0]	Reserved
RdCrcErr	[55:39]	Reserved
	[39:4]	Reserved
LinkEccErr	[55:39]	Reserved
	[39:4]	Reserved

**MSR0000\_043F...MSRC000\_2123 [UMC Machine Check Miscellaneous 0] (MCA::UMC::MCA\_MISC0\_UMC)**

Log miscellaneous information associated with errors.

\_instUMC\_n0\_umc0\_aliasMSRLEGACY; MSR0000\_043F

\_instUMC\_n1\_umc1\_aliasMSRLEGACY; MSR0000\_0443

\_instUMC\_n2\_umc0\_aliasMSRLEGACY; MSR0000\_0447

\_instUMC\_n3\_umc1\_aliasMSRLEGACY; MSR0000\_044B

\_instUMC\_n0\_umc0\_aliasMSR; MSRC000\_20F3

\_instUMC\_n1\_umc1\_aliasMSR; MSRC000\_2103

\_instUMC\_n2\_umc0\_aliasMSR; MSRC000\_2113

\_instUMC\_n3\_umc1\_aliasMSR; MSRC000\_2123

Bits	Description
63	<b>Valid.</b> Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	<b>CntP.</b> Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	<b>Locked.</b> Reset: 0. 1=Writes to this register are ignored. This bit is set by BIOS to indicate that this register is not available for OS use. BIOS should set this bit if ThresholdIntType is set to SMI. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
60	<b>IntP.</b> Reset: 1. 1=ThresholdIntType can be used to generate interrupts. 0=ThresholdIntType and interrupt generation are not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::UMC::MCA_MISC0_UMC[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	<b>LvtOffset.</b> Reset: 0h. One per die. For error thresholding interrupts, specifies the address of the LVT entry in the APIC registers as follows: LVT address = (LvtOffset shifted left 4 bits) + 500h (see Core::X86::Apic::ExtendedInterruptLvtEntries). AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::UMC::MCA_MISC0_UMC[Locked]) ? Read-write : Read-only.
51	<b>CntEn.</b> Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::UMC::MCA_MISC0_UMC[Locked]) ? Read-write : Read-only.
50:49	<b>ThresholdIntType.</b> Reset: Cold,0h. Specifies the type of interrupt signaled when Ovrflw is set and IntP==1. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[ThresholdLvtOffset]). 10b = SMI trigger event. 11b = Reserved. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::UMC::MCA_MISC0_UMC[Locked]) ? Read-write : Read-only.
48	<b>Ovrflw.</b> Reset: Cold,0. Set by hardware when ErrCnt transitions from FFEh to FFFh. When this field is set, ErrCnt no longer increments. When this bit is set, the interrupt selected by the ThresholdIntType field is generated. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::UMC::MCA_MISC0_UMC[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	<b>ErrCnt.</b> Reset: Cold,000h. This is written by software to set the starting value of the error counter. This is incremented by hardware when errors are logged. When this counter overflows, it stays at FFFh (no rollover). The threshold value, written by software, is (FFFh - the desired error count (the number of errors necessary in order for an interrupt to be taken)); the desired error count of 0 (a write value of FFFh) is not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::UMC::MCA_MISC0_UMC[Locked]) ? Read-write : Read-only.
31:24	<b>BlkPtr.</b> Read-write. Reset: 01h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

**MSRC000\_2[0F...12]4 [UMC Machine Check Configuration] (MCA::UMC::MCA\_CONFIG\_UMC)**

Reset: 0000_0000_0000_0125h.	
Controls configuration of the associated machine check bank.	
_instUMC_n0_umc0_aliasMSR; MSRC000_20F4	
_instUMC_n1_umc1_aliasMSR; MSRC000_2104	
_instUMC_n2_umc0_aliasMSR; MSRC000_2114	
_instUMC_n3_umc1_aliasMSR; MSRC000_2124	
Bits	Description
63:41	Reserved.
40	<b>IntEn.</b> Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	<b>DeferredIntType.</b> Read-write. Reset: 0h. Specifies the type of interrupt signaled when a deferred error is logged. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[DeferredLvtOffset]). 10b = SMI trigger event. 11b = Reserved.
36:35	Reserved.
34	<b>LogDeferredInMcaStat.</b> Read-write. Reset: 0. Init: BIOS,1. 1=Log deferred errors in MCA::UMC::MCA_STATUS_UMC and MCA::UMC::MCA_ADDR_UMC in addition to MCA::UMC::MCA_DESTAT_UMC and MCA::UMC::MCA_DEADDR_UMC. 0=Only log deferred errors in MCA::UMC::MCA_DESTAT_UMC and MCA::UMC::MCA_DEADDR_UMC. This bit does not affect logging of deferred errors in MCA::UMC::MCA_SYND_UMC, MCA::UMC::MCA_MISC0_UMC.
33	Reserved.
32	<b>McaXEnable.</b> Read-write. Reset: 0. Init: BIOS,1. Check: 1. 1=Software has acknowledged support for the MCAX feature set. 0=Software has not acknowledged support for the MCAX feature set. All uncorrected and fatal errors will cause an ErrorEvent packet to be generated. Deferred error interrupts are configured via Core::X86::Msr::McaIntrCfg.
31:11	Reserved.
10	<b>IntPresent.</b> Read-only, Volatile. Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::UMC::MCA_CONFIG_UMC[IntEn].
9	<b>McaFruTextInMca.</b> Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	<b>McaLsbInStatusSupported.</b> Read-only. Reset: 1. 1=MCA::UMC::MCA_CONFIG_UMC[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	<b>DeferredIntTypeSupported.</b> Read-only. Reset: 1. 1=MCA::UMC::MCA_CONFIG_UMC[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::UMC::MCA_CONFIG_UMC[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	<b>DeferredErrorLoggingSupported.</b> Read-only. Reset: 1. 1=Deferred errors are supported in this MCA bank, and MCA::UMC::MCA_CONFIG_UMC[LogDeferredInMcaStat] controls the logging behavior of these errors. MCA::UMC::MCA_DESTAT_UMC and MCA::UMC::MCA_DEADDR_UMC are supported in this MCA bank. 0=Deferred errors are not supported in this bank.
1	Reserved.
0	<b>McaX.</b> Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::UMC::MCA_MISC0_UMC[BlkPtr] indicates the presence of the additional MISC registers, but is not used to determine their MSR numbers. Deferred error interrupt type is specifiable by MCA bank. MCA::UMC::MCA_STATUS_UMC[TCC] is present.



**MSRC000\_2[0F...12]5 [UMC IP Identification] (MCA::UMC::MCA\_IPID\_UMC)**

Reset: 0000\_0096\_0000\_0000h.

The MCA::UMC::MCA\_IPID\_UMC register is used by software to determine what IP type and revision is associated with the MCA bank.

\_instUMC\_n0\_umc0\_aliasMSR; MSRC000\_20F5

\_instUMC\_n1\_umc1\_aliasMSR; MSRC000\_2105

\_instUMC\_n2\_umc0\_aliasMSR; MSRC000\_2115

\_instUMC\_n3\_umc1\_aliasMSR; MSRC000\_2125

Bits	Description
63:48	<b>McaType.</b> Read-only. Reset: 0000h. The McaType of the MCA bank within this IP.
47:44	<b>InstanceIdHi.</b> Read-write. Reset: 0h. The HI value instance ID of this IP. This is initialized to a unique ID per instance of this register.
43:32	<b>HardwareID.</b> Read-only. Reset: 096h. The Hardware ID of the IP associated with this MCA bank.
31:0	<b>InstanceId.</b> Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _instUMC_n0_umc0_aliasMSR: PSP,0005_0F00h
	Init: _instUMC_n1_umc1_aliasMSR: PSP,0015_0F00h
	Init: _instUMC_n2_umc0_aliasMSR: PSP,0025_0F00h
	Init: _instUMC_n3_umc1_aliasMSR: PSP,0035_0F00h

**MSRC000\_2[0F...12]6 [UMC Machine Check Syndrome] (MCA::UMC::MCA\_SYND\_UMC)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::UMC::MCA\_STATUS\_UMC Thread 0

\_instUMC\_n0\_umc0\_aliasMSR; MSRC000\_20F6

\_instUMC\_n1\_umc1\_aliasMSR; MSRC000\_2106

\_instUMC\_n2\_umc0\_aliasMSR; MSRC000\_2116

\_instUMC\_n3\_umc1\_aliasMSR; MSRC000\_2126

Bits	Description
63:32	<b>Syndrom.</b> Read-write, Volatile. Reset: Cold, 0000_0000h. Contains the syndrome, if any, associated with the error logged in MCA::UMC::MCA_STATUS_UMC. The low-order bit of the syndrome is stored in bit 0, and the syndrome has a length specified by MCA::UMC::MCA_SYND_UMC[Length]. The Syndrome field is only valid when MCA::UMC::MCA_SYND_UMC[Length] is not 0.
31:27	Reserved.
26:24	<b>ErrorPriority.</b> Read-write, Volatile. Reset: Cold, 0h. Encodes the priority of the error logged in MCA::UMC::MCA_SYND_UMC. 3'b000 = No error; 3'b001 = Reserved; 3'b010 = Corrected Error; 3'b011 = Deferred Error; 3'b100 = Uncorrected Error; 3'b101 = Fatal Error; all others reserved.
23:18	<b>Length.</b> Read-write, Volatile. Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::UMC::MCA_SYND_UMC[Syndrome]. A value of 0 indicates that there is no valid syndrome in MCA::UMC::MCA_SYND_UMC. For example, a syndrome length of 9 means that MCA::UMC::MCA_SYND_UMC[Syndrome] bits [8:0] contains a valid syndrome.
17:0	<b>ErrorInformation.</b> Read-write, Volatile. Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 66 [MCA_SYND_UMC].

Table 66: MCA\_SYND\_UMC

Error Type	Bits	Description
DramEccErr	[17:16]	Reserved
	[15]	Software-Managed Bad Symbol ID Error
	[14]	Reserved
	[13:8]	Symbol. Only contains valid information on a corrected error.
	[7:4]	Cid. Specifies the rank multiply ID for supported DIMMs.
	[3]	Sub channel
	[2:0]	Chip Select



WriteDataPoisonErr	[17:0]	Reserved
SdpParityErr	[17:0]	Reserved
ApbErr	[17:0]	Reserved
AddressCommandParityErr	[17:0]	Reserved
WriteDataCrcErr	[17:0]	Reserved
DcqSramEccErr	[17:14] [13:0]	Reserved Reserved
AesSramEccErr	[17] [16:8] [7:4] [3:2] [1:0]	Reserved Reserved Reserved Reserved Reserved
EcsRowErr	[17:0]	Reserved
EcsErr	[17:0]	Reserved
ThrttlErr	[17:0]	Reserved
RdCrcErr	[17:8] [7:4] [3] [2:0]	Reserved Cid. Specifies the rank multiply ID for supported DIMMs. Sub channel Chip Select
LinkEccErr	[17:16] [15] [14] [13:8] [7] [6:4] [3] [2:0]	Reserved Software-Managed Bad Symbol ID Error Reserved Symbol. Only contains valid information for corrected errors. Reserved Cid. Specifies the rank multiply ID for supported DIMMs. Reserved Chip Select

#### MSRC000\_2[0F...12]E [UMC Machine Check Syndrome Extended] (MCA::UMC::MCA\_SYND1\_UMC)

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::UMC::MCA\_STATUS\_UMC Thread 0

\_instUMC\_n0\_umc0\_aliasMSR; MSRC000\_20FE

\_instUMC\_n1\_umc1\_aliasMSR; MSRC000\_210E

\_instUMC\_n2\_umc0\_aliasMSR; MSRC000\_211E

\_instUMC\_n3\_umc1\_aliasMSR; MSRC000\_212E

Bits	Description
63:0	<b>Syndrome.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::UMC::MCA_SYND1_UMC register stores information associated with the error in MCA::UMC::MCA_STATUS_UMC or MCA_DESTAT. The register is meaningful if MCA::UMC::MCA_STATUS_UMC[SyndV]=1. When MCA::UMC::MCA_CONFIG_UMC[McaFruTextInMca]=1, MCA::UMC::MCA_SYND1_UMC stores ASCII FruText associated with the error.

**MSRC000\_2[0F...12]F [UMC Machine Check Syndrome Extended] (MCA::UMC::MCA\_SYND2\_UMC)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::UMC::MCA\_STATUS\_UMC Thread 0

\_instUMC\_n0\_umc0\_aliasMSR; MSRC000\_20FF\_instUMC\_n1\_umc1\_aliasMSR; MSRC000\_210F\_instUMC\_n2\_umc0\_aliasMSR; MSRC000\_211F\_instUMC\_n3\_umc1\_aliasMSR; MSRC000\_212F

Bits	Description
63:0	<b>Syndrome.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::UMC::MCA_SYND2_UMC register stores information associated with the error in MCA::UMC::MCA_STATUS_UMC or MCA_DESTAT. The register is meaningful if MCA::UMC::MCA_STATUS_UMC[SyndV]=1. When MCA::UMC::MCA_CONFIG_UMC[McaFruTextInMca]=1, MCA::UMC::MCA_SYND2_UMC stores ASCII FruText associated with the error.

**MSRC000\_2[0F...12]8 [UMC Machine Check Deferred Error Status] (MCA::UMC::MCA\_DESTAT\_UMC)**

Reset: Cold, 0000\_0000\_0000\_0000h.

Holds status information for the first deferred error seen in this bank.

\_instUMC\_n0\_umc0\_aliasMSR; MSRC000\_20F8\_instUMC\_n1\_umc1\_aliasMSR; MSRC000\_2108\_instUMC\_n2\_umc0\_aliasMSR; MSRC000\_2118\_instUMC\_n3\_umc1\_aliasMSR; MSRC000\_2128

Bits	Description
63	<b>Val.</b> Read-write, Volatile. Reset: Cold, 0. 1=A valid error has been detected (whether it is enabled or not).
62	<b>Overflow.</b> Read-write, Volatile. Reset: Cold, 0. 1=An error was detected while the valid bit (Val) was set; at least one error was not logged. Overflow is set independently of whether the existing error is overwritten. (See the section on error overwrite priorities.)
61:59	<b>RESERV4.</b> Read-write. Reset: Cold, 0h. MCA_DEFSTAT Register Reserved bits.
58	<b>AddrV.</b> Read-write, Volatile. Reset: Cold, 0. 1=MCA::UMC::MCA_DEADDR_UMC contains address information associated with the error.
57:54	<b>RESERV3.</b> Read-write. Reset: Cold, 0h. MCA_DEFSTAT Register Reserved bits.
53	<b>SyndV.</b> Read-write, Volatile. Reset: Cold, 0. 1=This error logged information in MCA::UMC::MCA_SYND_UMC. If MCA::UMC::MCA_SYND_UMC[ErrorPriority] is the same as the priority of the error in MCA::UMC::MCA_STATUS_UMC, then the information in MCA::UMC::MCA_SYND_UMC is associated with the error in MCA::UMC::MCA_DESTAT_UMC.
52:45	<b>RESERV2.</b> Read-write. Reset: Cold, 00h. MCA_DEFSTAT Register Reserved bits.
44	<b>Deferred.</b> Read-write, Volatile. Reset: Cold, 0. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; poison is created and an exception is deferred until the poison data is consumed.
43:30	<b>RESERV1.</b> Read-write. Reset: Cold, 0000h. MCA_DEFSTAT Register Reserved bits.
29:24	<b>AddrLsb.</b> Read-write, Volatile. Reset: Cold, 00h. Specifies the least significant valid bit of the address contained in MCA::UMC::MCA_ADDR_UMC[ErrorAddr]. A value of 0 indicates that MCA::UMC::MCA_ADDR_UMC[63:0] contains a valid byte address. A value of 6 indicates that MCA::UMC::MCA_ADDR_UMC[63:6] contains a valid cache line address and that MCA::UMC::MCA_ADDR_UMC[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::UMC::MCA_ADDR_UMC[63:12] contain a valid 4KB memory page and that MCA::UMC::MCA_ADDR_UMC[11:0] should be ignored by error handling software.
23:22	<b>RESERV0.</b> Read-write. Reset: Cold, 0h. MCA_DEFSTAT Register Reserved bits.
21:16	<b>ErrorCodeExt.</b> Read-write, Volatile. Reset: Cold, 00h. Logs an extended error code when an error is detected. This model-specific field is used in conjunction with ErrorCode to identify the error sub-type for root cause analysis.
15:0	<b>ErrorCode.</b> Read-write, Volatile. Reset: Cold, 0000h. Error code for this error.

**MSRC000\_2[0F...12]9 [UMC Deferred Error Address] (MCA::UMC::MCA\_DEADDR\_UMC)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

The MCA::UMC::MCA\_DEADDR\_UMC register stores the address associated with the error in MCA::UMC::MCA\_DESTAT\_UMC. The register is only meaningful if MCA::UMC::MCA\_DESTAT\_UMC[Val]=1 and MCA::UMC::MCA\_DESTAT\_UMC[AddrV]=1. The lowest valid bit of the address is defined by MCA::UMC::MCA\_DESTAT\_UMC[AddrLsb].

\_instUMC\_n0\_umc0\_aliasMSR; MSRC000\_20F9

\_instUMC\_n1\_umc1\_aliasMSR; MSRC000\_2109

\_instUMC\_n2\_umc0\_aliasMSR; MSRC000\_2119

\_instUMC\_n3\_umc1\_aliasMSR; MSRC000\_2129

Bits	Description
63:0	<b>ErrorAddr.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::UMC::MCA_DESTAT_UMC. The lowest-order valid bit of the address is specified in MCA::UMC::MCA_DESTAT_UMC[AddrLsb].

**MSRC000\_2[0F...12]A [UMC Machine Check Miscellaneous 1] (MCA::UMC::MCA\_MISC1\_UMC)**

Log miscellaneous information associated with errors, as defined by each error type.

\_instUMC\_n0\_umc0\_aliasMSR; MSRC000\_20FA

\_instUMC\_n1\_umc1\_aliasMSR; MSRC000\_210A

\_instUMC\_n2\_umc0\_aliasMSR; MSRC000\_211A

\_instUMC\_n3\_umc1\_aliasMSR; MSRC000\_212A

Bits	Description
63	<b>Valid.</b> Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	<b>CntP.</b> Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	<b>Locked.</b> Reset: 0. 1=Writes to this register are ignored. This bit is set by BIOS to indicate that this register is not available for OS use. BIOS should set this bit if ThresholdIntType is set to SMI. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
60	<b>IntP.</b> Reset: 1. 1=ThresholdIntType can be used to generate interrupts. 0=ThresholdIntType and interrupt generation are not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::UMC::MCA_MISC1_UMC[Locked]) ? Read-write : Read-only.
59:52	Reserved.
51	<b>CntEn.</b> Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::UMC::MCA_MISC1_UMC[Locked]) ? Read-write : Read-only.
50:49	<b>ThresholdIntType.</b> Reset: Cold,0h. Specifies the type of interrupt signaled when Ovrflw is set. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[ThresholdLvtOffset]) to all cores. 10b = SMI trigger event. 11b = Reserved. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::UMC::MCA_MISC1_UMC[Locked]) ? Read-write : Read-only.
48	<b>Ovrflw.</b> Reset: Cold,0. Set by hardware when ErrCnt transitions from FFEh to FFFh; also set by hardware if ErrCnt is initialized to FFFh and transitions from FFFh to 000h. When this field is set, ErrCnt no longer increments. When this bit is set, the interrupt selected by the ThresholdIntType field is generated. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::UMC::MCA_MISC1_UMC[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	<b>ErrCnt.</b> Reset: Cold,000h. This is written by software to set the starting value of the error counter. This is incremented by hardware when errors are logged. When this counter overflows, it stays at FFFh (no rollover). The threshold value, written by software, is (FFFh - the desired error count (the number of errors necessary in order for an interrupt to be taken)); the desired error count of 0 (a write value of FFFh) is not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::UMC::MCA_MISC1_UMC[Locked]) ? Read-write : Read-only.
31:24	<b>BlkPtr.</b> Read-write. Reset: 01h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

**MSRC001\_04[0F...12] [UMC Machine Check Control Mask] (MCA::UMC::MCA\_CTL\_MASK\_UMC)**

Read-write. Reset: 0000_0000_0000_0000h.	
Inhibit detection of an error source.	
_instUMC_n0_umc0_aliasMSR; MSRC001_040F	
_instUMC_n1_umc1_aliasMSR; MSRC001_0410	
_instUMC_n2_umc0_aliasMSR; MSRC001_0411	
_instUMC_n3_umc1_aliasMSR; MSRC001_0412	
Bits	Description
63:13	Reserved.
12	<b>LinkEccErr.</b> Read-write. Reset: 0. Link ECC error. An ECC error occurred on a DRAM link
11	<b>RdCrcErr.</b> Read-write. Reset: 0. Read CRC error. CRC error occurred on a DRAM read from any subchannel
10	<b>ThrttlErr.</b> Read-write. Reset: 0. Indicates that UMC is throttling
9	<b>EcsErr.</b> Read-write. Reset: 0. ECS Error. Indicates that a device exceeded the ECS Error Threshold Count.
8	<b>EcsRowErr.</b> Read-write. Reset: 0. ECS Row Error. Indicates that a single device row exceeded four code word errors.
7	<b>AesSramEccErr.</b> Read-write. Reset: 0. AES SRAM ECC error. An ECC error occurred on a AES SRAM in the processor.
6	<b>DcqSramEccErr.</b> Read-write. Reset: 0. DCQ SRAM ECC error. An ECC error occurred on a DCQ SRAM in the processor.
5	<b>WriteDataCrcErr.</b> Read-write. Reset: 0. Write data CRC error. A write data CRC error occurred on the DRAM data bus.
4	<b>AddressCommandParityErr.</b> Read-write. Reset: 0. Address/Command parity error. A parity error occurred on the DRAM address/command bus.
3	<b>ApbErr.</b> Read-write. Reset: 0. Advanced peripheral bus error. An error occurred on the advanced peripheral bus.
2	<b>SdpParityErr.</b> Read-write. Reset: 0. SDP parity error. A parity error was detected on write data from the data fabric in the processor.
1	<b>WriteDataPoisonErr.</b> Read-write. Reset: 0. Data poison error. The system tried to write poison data to DRAM and either DRAM does not support ECC or UMC_CH.EccCtrl.WrEccEn is cleared.
0	<b>DramEccErr.</b> Read-write. Reset: 0. DRAM ECC error. An ECC error occurred on a DRAM read.

**3.2.5.11 NBIO****MSR0000\_0470...MSRC000\_21C0 [NBIO Machine Check Control] (MCA::NBIO::MCA\_CTL\_NBIO)**

Read-write. Reset: 0000_0000_0000_0000h.	
0=Disables error reporting for the corresponding error. 1=Enables error reporting via machine check exception for the corresponding error. The MCA::NBIO::MCA_CTL_NBIO register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG_CTL. Does not affect error detection, correction, or logging.	
_instIOHC_n0_aliasMSRLEGACY; MSR0000_0470	
_instIOHC_n0_aliasMSR; MSRC000_21C0	
Bits	Description
63:6	Reserved.
5	<b>Int_ErrEvent.</b> Read-write. Reset: 0. Internal system fatal error event was detected.
4	<b>IOHC_Internal_Poison.</b> Read-write. Reset: 0. Internal Poison Error. Poison data was sent to an internal client.
3	<b>Egress_Poison.</b> Read-write. Reset: 0. SDP Egress Poison Error. Poison was propagated to an egress port.
2	<b>Ext_ErrEvent.</b> Read-write. Reset: 0. External SDP ErrEvent error. A system fatal error event from an SDP interface was detected.
1	<b>PCIE_Sideband.</b> Read-write. Reset: 0. PCIe error. A PCIe® error was logged in a PCIe® root port.
0	<b>EccParityError.</b> Read-write. Reset: 0. ECC or Parity error. An SRAM ECC or parity error was detected.

**MSR0000\_0471...MSRC000\_21C1 [NBIO Machine Check Status] (MCA::NBIO::MCA\_STATUS\_NBIO)**

Reset: Cold,0000\_0000\_0000\_0000h.

Logs information associated with errors.

\_instIOHC\_n0\_aliasMSRLEGACY; MSR0000\_0471

\_instIOHC\_n0\_aliasMSR; MSRC000\_21C1

Bits	Description
63	<b>Val.</b> Reset: Cold,0. 1=A valid error has been detected. This bit should be cleared by software after the register has been read. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
62	<b>Overflow.</b> Reset: Cold,0. 1=An error was detected while the valid bit (Val) was set; at least one error was not logged. Overflow is set independently of whether the existing error is overwritten. See 3.1.3 [Machine Check Errors]. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
61	<b>UC.</b> Reset: Cold,0. 1=The error was not corrected by hardware. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
60	<b>En.</b> Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::NBIO::MCA_CTL_NBIO. This bit is a copy of bit in MCA::NBIO::MCA_CTL_NBIO for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	<b>MiscV.</b> Reset: Cold,0. 1=Valid thresholding in MCA::NBIO::MCA_MISC0_NBIO. In certain modes, MISC registers are owned by platform firmware and will RAZ when read by non-SMM code. Therefore, it is possible for MiscV=1 and the MISC register to read as all zeros. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
58	<b>AddrV.</b> Reset: Cold,0. 1=MCA::NBIO::MCA_ADDR_NBIO contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	<b>PCC.</b> Reset: Cold,0. 1=Hardware context held by the processor may have been corrupted. Continued operation of the system may have unpredictable results. The error is not recoverable or survivable, and the system should be reinitialized. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
56	<b>ErrCoreIdVal.</b> Reset: Cold,0. 1=The ErrCoreId field is valid. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
55	<b>TCC.</b> Reset: Cold,0. 1=Hardware context of the process thread to which the error was reported may have been corrupted. Continued operation of the thread may have unpredictable results. The thread must be terminated. Only meaningful when MCA::NBIO::MCA_STATUS_NBIO[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	<b>RESERV54.</b> Reset: Cold,0. MCA_STATUS Register Reserved bit. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
53	<b>SyndV.</b> Reset: Cold,0. 1=This error logged information in MCA::NBIO::MCA_SYND_NBIO. If MCA::NBIO::MCA_SYND_NBIO[ErrorPriority] is the same as the priority of the error in MCA::NBIO::MCA_STATUS_NBIO, then the information in MCA::NBIO::MCA_SYND_NBIO is associated with the error in MCA::NBIO::MCA_STATUS_NBIO. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	<b>RESERV47.</b> Reset: Cold,00h. MCA_STATUS Register Reserved bits. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
46	<b>CECC.</b> Reset: Cold,0. 1=The error was a correctable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
45	<b>UECC.</b> Reset: Cold,0. 1=The error was an uncorrectable ECC error according to the restrictions of the ECC algorithm. UC indicates whether the error was actually corrected by the processor.



	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
44	<b>Deferred.</b> Reset: Cold,0. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; an exception is deferred until the erroneous data is consumed.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
43	<b>Poison.</b> Reset: Cold,0. 1=The error was the result of attempting to consume poisoned data.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
42:41	<b>RESERV41.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
40	<b>Scrub.</b> Reset: Cold,0. 1=The error was the result of a scrub operation.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
39:38	<b>RESERV38.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
37:32	<b>ErrCoreId.</b> Reset: Cold,00h. When ErrCoreIdVal=1 this field indicates which core within the processor is associated with the error; Otherwise this field is reserved.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
31:30	<b>RESERV30.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
29:24	<b>AddrLsb.</b> Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::NBIO::MCA_ADDR_NBIO[ErrorAddr]. A value of 0 indicates that MCA::NBIO::MCA_ADDR_NBIO[63:0] contains a valid byte address. A value of 6 indicates that MCA::NBIO::MCA_ADDR_NBIO[63:6] contains a valid cache line address and that MCA::NBIO::MCA_ADDR_NBIO[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::NBIO::MCA_ADDR_NBIO[63:12] contain a valid 4KB memory page and that MCA::NBIO::MCA_ADDR_NBIO[11:0] should be ignored by error handling software.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
23:22	<b>RESERV22.</b> Reset: Cold,0h. MCA_STATUS Register Reserved bits.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
21:16	<b>ErrorCodeExt.</b> Reset: Cold,00h. Extended Error Code. This field is used to identify the error type for root cause analysis. This field indicates which bit position in MCA::NBIO::MCA_CTL_NBIO enables error reporting for the logged error.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
15:0	<b>ErrorCode.</b> Reset: Cold,0000h. Error code for this error. See 3.1.3.3 [Error Codes] for details on decoding this field.
	AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.

Table 67: MCA\_STATUS\_NBIO

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
EccParityError	0x0	0/1	0/1	0/1	0/1	0	0
PCIE_Sideband	0x1	0/1	0/1	0/1	0/1	0	0
Ext_ErrEvent	0x2	1	1	1	0	0	0
Egress_Poison	0x3	0/1	0/1	0/1	0/1	0	0
IOHC_Internal_Poison	0x4	1	1	1	0	0	0
Int_ErrEvent	0x5	1	1	1	0	0	0

**MSR0000\_0472...MSRC000\_21C2 [NBIO Machine Check Address] (MCA::NBIO::MCA\_ADDR\_NBIO)**

Read-only. Reset: Cold,0000\_0000\_0000\_0000h.

MCA::NBIO::MCA\_ADDR\_NBIO stores an address and other information associated with the error in MCA::NBIO::MCA\_STATUS\_NBIO. The register is only meaningful if MCA::NBIO::MCA\_STATUS\_NBIO[Val]=1 and MCA::NBIO::MCA\_STATUS\_NBIO[AddrV]=1.

\_instIOHC\_n0\_aliasMSRLEGACY; MSR0000\_0472

\_instIOHC\_n0\_aliasMSR; MSRC000\_21C2

Bits	Description
63:0	<b>ErrorAddr.</b> Read-only. Reset: Cold,0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::NBIO::MCA_STATUS_NBIO.

Table 68: MCA\_ADDR\_NBIO

Error Type	Bits	Description
EccParityError	[63:0]	Reserved
PCIE_Sideband	[63:0]	Reserved
Ext_ErrEvent	[63:0]	Reserved
Egress_Poison	[63:0]	Reserved
IOHC_Internal_Poison	[63:0]	Reserved
Int_ErrEvent	[63:0]	Reserved



**MSR0000\_0473...MSRC000\_21C3 [NBIO Machine Check Miscellaneous 0] (MCA::NBIO::MCA\_MISC0\_NBIO)**

Log miscellaneous information associated with errors.

\_instIOHC\_n0\_aliasMSRLEGACY; MSR0000\_0473

\_instIOHC\_n0\_aliasMSR; MSRC000\_21C3

Bits	Description
63	<b>Valid.</b> Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	<b>CntP.</b> Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	<b>Locked.</b> Reset: 0. 1=Writes to this register are ignored. This bit is set by BIOS to indicate that this register is not available for OS use. BIOS should set this bit if ThresholdIntType is set to SMI. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
60	<b>IntP.</b> Reset: 1. 1=ThresholdIntType can be used to generate interrupts. 0=ThresholdIntType and interrupt generation are not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::NBIO::MCA_MISC0_NBIO[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	<b>LvtOffset.</b> Reset: 0h. One per die. For error thresholding interrupts, specifies the address of the LVT entry in the APIC registers as follows: LVT address = (LvtOffset shifted left 4 bits) + 500h (see Core::X86::Apic::ExtendedInterruptLvtEntries). AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::NBIO::MCA_MISC0_NBIO[Locked]) ? Read-write : Read-only.
51	<b>CntEn.</b> Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::NBIO::MCA_MISC0_NBIO[Locked]) ? Read-write : Read-only.
50:49	<b>ThresholdIntType.</b> Reset: Cold,0h. Specifies the type of interrupt signaled when Ovrlw is set and IntP==1. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[ThresholdLvtOffset]). 10b = SMI trigger event. 11b = Reserved. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::NBIO::MCA_MISC0_NBIO[Locked]) ? Read-write : Read-only.
48	<b>Ovrlw.</b> Reset: Cold,0. Set by hardware when ErrCnt transitions from FFEh to FFFh. When this field is set, ErrCnt no longer increments. When this bit is set, the interrupt selected by the ThresholdIntType field is generated. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::NBIO::MCA_MISC0_NBIO[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	<b>ErrCnt.</b> Reset: Cold,000h. This is written by software to set the starting value of the error counter. This is incremented by hardware when errors are logged. When this counter overflows, it stays at FFFh (no rollover). The threshold value, written by software, is (FFFh - the desired error count (the number of errors necessary in order for an interrupt to be taken)); the desired error count of 0 (a write value of FFFh) is not supported. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn]   !MCA::NBIO::MCA_MISC0_NBIO[Locked]) ? Read-write : Read-only.
31:24	<b>BlkPtr.</b> Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

**MSRC000\_21C4 [NBIO Machine Check Configuration] (MCA::NBIO::MCA\_CONFIG\_NBIO)**

Reset: 0000_0002_0000_0125h.	
Controls configuration of the associated machine check bank.	
_instIOHC_n0_aliasMSR; MSRC000_21C4	
Bits	Description
63:41	Reserved.
40	<b>IntEn.</b> Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	<b>DeferredIntType.</b> Read-write. Reset: 0h. Specifies the type of interrupt signaled when a deferred error is logged. 00b = No interrupt. 01b = APIC based interrupt (see Core::X86::Msr::McaIntrCfg[DeferredLvtOffset]). 10b = SMI trigger event. 11b = Reserved.
36:35	Reserved.
34	<b>LogDeferredInMcaStat.</b> Read-write. Reset: 0. Init: BIOS,1. 1=Log deferred errors in MCA::NBIO::MCA_STATUS_NBIO and MCA::NBIO::MCA_ADDR_NBIO in addition to MCA::NBIO::MCA_DESTAT_NBIO and MCA::NBIO::MCA_DEADDR_NBIO. 0=Only log deferred errors in MCA::NBIO::MCA_DESTAT_NBIO and MCA::NBIO::MCA_DEADDR_NBIO. This bit does not affect logging of deferred errors in MCA::NBIO::MCA_SYND_NBIO, MCA::NBIO::MCA_MISC0_NBIO.
33	Reserved.
32	<b>McaXEnable.</b> Read-write. Reset: 0. Init: BIOS,1. Check: 1. 1=Software has acknowledged support for the MCAX feature set. 0=Software has not acknowledged support for the MCAX feature set. All uncorrected and fatal errors will cause an ErrorEvent packet to be generated. Deferred error interrupts are configured via Core::X86::Msr::McaIntrCfg.
31:11	Reserved.
10	<b>IntPresent.</b> Read-only, Volatile. Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::NBIO::MCA_CONFIG_NBIO[IntEn].
9	<b>McaFruTextInMca.</b> Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	<b>McaLsbInStatusSupported.</b> Read-only. Reset: 1. 1=MCA::NBIO::MCA_CONFIG_NBIO[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	<b>DeferredIntTypeSupported.</b> Read-only. Reset: 1. 1=MCA::NBIO::MCA_CONFIG_NBIO[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::NBIO::MCA_CONFIG_NBIO[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	<b>DeferredErrorLoggingSupported.</b> Read-only. Reset: 1. 1=Deferred errors are supported in this MCA bank, and MCA::NBIO::MCA_CONFIG_NBIO[LogDeferredInMcaStat] controls the logging behavior of these errors. MCA::NBIO::MCA_DESTAT_NBIO and MCA::NBIO::MCA_DEADDR_NBIO are supported in this MCA bank. 0=Deferred errors are not supported in this bank.
1	Reserved.
0	<b>McaX.</b> Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::NBIO::MCA_MISC0_NBIO[BlkPtr] indicates the presence of the additional MISC registers, but is not used to determine their MSR numbers. Deferred error interrupt type is specifiable by MCA bank. MCA::NBIO::MCA_STATUS_NBIO[TCC] is present.

**MSRC000\_21C5 [NBIO IP Identification] (MCA::NBIO::MCA\_IPID\_NBIO)**

Reset: 0000\_0018\_0000\_0000h.

The MCA::NBIO::MCA\_IPID\_NBIO register is used by software to determine what IP type and revision is associated with the MCA bank.

\_instIOHC\_n0\_aliasMSR; MSRC000\_21C5

Bits	Description
63:48	<b>McaType.</b> Read-only. Reset: 0000h. The McaType of the MCA bank within this IP.
47:44	<b>InstanceIdHi.</b> Read-write. Reset: 0h. The HI value instance ID of this IP. This is initialized to a unique ID per instance of this register.
43:32	<b>HardwareID.</b> Read-only. Reset: 018h. The Hardware ID of the IP associated with this MCA bank.
31:0	<b>InstanceId.</b> Read-write. Reset: 0000_0000h. Init: PSP,13B1_7000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.

**MSRC000\_21C6 [NBIO Machine Check Syndrome] (MCA::NBIO::MCA\_SYND\_NBIO)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::NBIO::MCA\_STATUS\_NBIO Thread 0

\_instIOHC\_n0\_aliasMSR; MSRC000\_21C6

Bits	Description
63:33	Reserved.
32	<b>Syndrom.</b> Read-write, Volatile. Reset: Cold, 0. Contains the syndrome, if any, associated with the error logged in MCA::NBIO::MCA_STATUS_NBIO. The low-order bit of the syndrome is stored in bit 0, and the syndrome has a length specified by MCA::NBIO::MCA_SYND_NBIO[Length]. The Syndrome field is only valid when MCA::NBIO::MCA_SYND_NBIO[Length] is not 0.
31:27	Reserved.
26:24	<b>ErrorPriority.</b> Read-write, Volatile. Reset: Cold, 0h. Encodes the priority of the error logged in MCA::NBIO::MCA_SYND_NBIO. 3'b000 = No error; 3'b001 = Reserved; 3'b010 = Corrected Error; 3'b011 = Deferred Error; 3'b100 = Uncorrected Error; 3'b101 = Fatal Error; all others reserved.
23:18	<b>Length.</b> Read-write, Volatile. Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::NBIO::MCA_SYND_NBIO[Syndrome]. A value of 0 indicates that there is no valid syndrome in MCA::NBIO::MCA_SYND_NBIO. For example, a syndrome length of 9 means that MCA::NBIO::MCA_SYND_NBIO[Syndrome] bits [8:0] contains a valid syndrome.
17:0	<b>ErrorInformation.</b> Read-write, Volatile. Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 69 [MCA_SYND_NBIO].

Table 69: MCA\_SYND\_NBIO

Error Type	Bits	Description
EccParityError	[17:5] [4:0]	Group ID Structure ID
PCIE_Sideband	[5:0]	EgressPortNum
Ext_ErrEvent	[3:0]	Reserved
Egress_Poison	[5:0]	Egress Port Number
IOHC_Internal_Poison	[0]	0:CfgMaster 1:TrapClient
Int_ErrEvent	[0]	Reserved

**MSRC000\_21C8 [NBIO Machine Check Deferred Error Status] (MCA::NBIO::MCA\_DESTAT\_NBIO)**

Reset: Cold,0000\_0000\_0000\_0000h.

Holds status information for the first deferred error seen in this bank.

\_instIOHC\_n0\_aliasMSR; MSRC000\_21C8

Bits	Description
63	<b>Val.</b> Read-write, Volatile. Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	<b>Overflow.</b> Read-write, Volatile. Reset: Cold,0. 1=An error was detected while the valid bit (Val) was set; at least one error was not logged. Overflow is set independently of whether the existing error is overwritten. (See the section on error overwrite priorities.)
61:59	<b>RESERV4.</b> Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	<b>AddrV.</b> Read-write, Volatile. Reset: Cold,0. 1=MCA::NBIO::MCA_DEADDR_NBIO contains address information associated with the error.
57:54	<b>RESERV3.</b> Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	<b>SyndV.</b> Read-write, Volatile. Reset: Cold,0. 1=This error logged information in MCA::NBIO::MCA_SYND_NBIO. If MCA::NBIO::MCA_SYND_NBIO[ErrorPriority] is the same as the priority of the error in MCA::NBIO::MCA_STATUS_NBIO, then the information in MCA::NBIO::MCA_SYND_NBIO is associated with the error in MCA::NBIO::MCA_DESTAT_NBIO.
52:45	<b>RESERV2.</b> Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	<b>Deferred.</b> Read-write, Volatile. Reset: Cold,0. 1=A deferred error was created. A deferred error is the result of an uncorrectable data error which did not immediately cause a processor exception; poison is created and an exception is deferred until the poison data is consumed.
43:30	<b>RESERV1.</b> Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	<b>AddrLsb.</b> Read-write, Volatile. Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::NBIO::MCA_ADDR_NBIO[ErrorAddr]. A value of 0 indicates that MCA::NBIO::MCA_ADDR_NBIO[63:0] contains a valid byte address. A value of 6 indicates that MCA::NBIO::MCA_ADDR_NBIO[63:6] contains a valid cache line address and that MCA::NBIO::MCA_ADDR_NBIO[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::NBIO::MCA_ADDR_NBIO[63:12] contain a valid 4KB memory page and that MCA::NBIO::MCA_ADDR_NBIO[11:0] should be ignored by error handling software.
23:22	<b>RESERV0.</b> Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	<b>ErrorCodeExt.</b> Read-write, Volatile. Reset: Cold,00h. Logs an extended error code when an error is detected. This model-specific field is used in conjunction with ErrorCode to identify the error sub-type for root cause analysis.
15:0	<b>ErrorCode.</b> Read-write, Volatile. Reset: Cold,0000h. Error code for this error.

**MSRC000\_21C9 [NBIO Deferred Error Address] (MCA::NBIO::MCA\_DEADDR\_NBIO)**

Read-only. Reset: Cold,0000\_0000\_0000\_0000h.

The MCA::NBIO::MCA\_DEADDR\_NBIO register stores the address associated with the error in MCA::NBIO::MCA\_DESTAT\_NBIO. The register is only meaningful if MCA::NBIO::MCA\_DESTAT\_NBIO[Val]=1 and MCA::NBIO::MCA\_DESTAT\_NBIO[AddrV]=1. The lowest valid bit of the address is defined by MCA::NBIO::MCA\_DESTAT\_NBIO[AddrLsb].

\_instIOHC\_n0\_aliasMSR; MSRC000\_21C9

Bits	Description
63:0	<b>ErrorAddr.</b> Read-only. Reset: Cold,0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::NBIO::MCA_DESTAT_NBIO.

**MSRC001\_041C [NBIO Machine Check Control Mask] (MCA::NBIO::MCA\_CTL\_MASK\_NBIO)**

Read-write. Reset: 0000\_0000\_0000\_0000h.

Inhibit detection of an error source.

\_instIOHC\_n0\_aliasMSR; MSRC001\_041C

Bits	Description
63:6	Reserved.
5	<b>Int_ErrEvent.</b> Read-write. Reset: 0. Init: BIOS,1. Internal system fatal error event was detected.
4	<b>IOHC_Internal_Poison.</b> Read-write. Reset: 0. Internal Poison Error. Poison data was sent to an internal client.
3	<b>Egress_Poison.</b> Read-write. Reset: 0. SDP Egress Poison Error. Poison was propagated to an egress port.
2	<b>Ext_ErrEvent.</b> Read-write. Reset: 0. Init: BIOS,1. External SDP ErrEvent error. A system fatal error event from an SDP interface was detected.
1	<b>PCIE_Sideband.</b> Read-write. Reset: 0. Init: BIOS,1. PCIE error. A PCIe® error was logged in a PCIe® root port.
0	<b>EccParityError.</b> Read-write. Reset: 0. ECC or Parity error. An SRAM ECC or parity error was detected.

**MSRC000\_21CE [NBIO Machine Check Syndrome Extended] (MCA::NBIO::MCA\_SYND1\_NBIO)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::NBIO::MCA\_STATUS\_NBIO Thread 0

\_instIOHC\_n0\_aliasMSR; MSRC000\_21CE

Bits	Description
63:0	<b>Syndrome.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::NBIO::MCA_SYND1_NBIO register stores information associated with the error in MCA::NBIO::MCA_STATUS_NBIO or MCA_DESTAT. The register is meaningful if MCA::NBIO::MCA_STATUS_NBIO[SyndV]=1. When MCA::NBIO::MCA_CONFIG_NBIO[McaFruTextInMca]=1, MCA::NBIO::MCA_SYND1_NBIO stores ASCII FruText associated with the error.

**MSRC000\_21CF [NBIO Machine Check Syndrome Extended] (MCA::NBIO::MCA\_SYND2\_NBIO)**

Read-write, Volatile. Reset: Cold, 0000\_0000\_0000\_0000h.

Logs physical location information associated with error in MCA::NBIO::MCA\_STATUS\_NBIO Thread 0

\_instIOHC\_n0\_aliasMSR; MSRC000\_21CF

Bits	Description
63:0	<b>Syndrome.</b> Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h. The MCA::NBIO::MCA_SYND2_NBIO register stores information associated with the error in MCA::NBIO::MCA_STATUS_NBIO or MCA_DESTAT. The register is meaningful if MCA::NBIO::MCA_STATUS_NBIO[SyndV]=1. When MCA::NBIO::MCA_CONFIG_NBIO[McaFruTextInMca]=1, MCA::NBIO::MCA_SYND2_NBIO stores ASCII FruText associated with the error.

## 4 Advanced Platform Management Link (APML)

### 4.1 Overview

The Advanced Platform Management Link (APML) is a SMBus v2.0 compatible 2-wire processor slave interface. APML is also referred as the sideband interface (SBI).

APML is used to communicate with the Remote Management Interface (see SBI Remote Management Interface (SB-RMI) and SBI Temperature Sensor Interface (SB-TSI). For related specifications, see 1.2 [Reference Documents].

#### 4.1.1 Definitions

Table 70: APML Definitions

Term	Description
<b>ARA</b>	Alert response address.
<b>ARP</b>	Address Resolution Protocol
<b>EC</b>	Embedded Controller.
<b>KBC</b>	Keyboard Controller.
<b>Master or SMBus Master</b>	The device that initiates and terminates all communication and drives the clock, SCL.
<b>PEC</b>	Packet error code.
<b>POR</b>	Power on reset.
<b>RTS</b>	Remote temperature sensor, typical examples are ADM1032, LM99, MAX6657, EMC1002.
<b>SB-RMI</b>	Remote Management interface.
<b>Slave or SMBus slave</b>	The slave cannot initiate SMBus communication and cannot drive the clock but can drive the data signal SDA and the alert signal ALERT_L.
<b>TSI</b>	Temperature sensor interface.

### 4.2 SBI Bus Characteristics

The SBI largely follows SMBus v2.0. This section describes the exceptions.

#### 4.2.1 SMBus Protocol Support

The SBI follows SMBus protocol except:

- The processor does not implement SMBus master functionality.
- The SBI implements the Send Byte/Receive Byte, Read Byte/Write Byte, Block Read/Block Write and Block Write-Block Read Process Call SMBus protocols. The Send Byte/Receive Byte SMBus protocol is only supported by SB-TSI.
- Packet error checking (PEC) is not supported by SB-TSI.
- Address Resolution Protocol (ARP) is not implemented.
- Cumulative clock extensions are not enforced.

#### 4.2.2 I2C Support

The processor supports higher I2C-defined speeds as specified in the Physical Layer Characteristics section. The

processor supports the I2C master code transmission in order to reach the high-speed bus mode. Multiple SBI commands may be sent within a single high-speed mode session. Ten-bit addressing is not supported.

## 4.3 SBI Processor Information

### 4.3.1 SBI Processor Pins

Up to six processor pins are used for SBI support: two for data transfer, three for address determination and one for an interrupt output. Of the three address pins, one bit is `socket_id` used to determine which package is addressed. These pins do not have changeable pinstrap. The Serial Interface Clock (SIC) and Serial Interface Data (SID) pins function as the SMBus clock and data pins respectively. The SMBus alert pin (`ALERT_L`) is used to signal interrupts to the SMBus master.

#### 4.3.1.1 Physical Layer Characteristics

The SIC and SID pins differ from the SMBus specification with regard to voltage. System board voltage translators are necessary to convert the SIC and SID pin voltage levels to that of the SMBus specification. SBI supports frequencies of 100 KHz, 400 KHz over SIC.

### 4.3.2 Processor States

SBI responds to SMBus traffic except when `PWROK` is de-asserted (and for a brief period after it is de-asserted). Access to internal processor state using SB-RMI is not supported under the following conditions:

- During cold and warm resets.
- During the APIC spin loop.

## 4.4 SBI Protocols

### 4.4.1 SBI Modified Block Write-Block Read Process Call

SBI uses a modified SMBus PEC-optional Block Write-Block Read Process Call protocol. The change from the SMBus protocol is support for an optional intermediate PEC byte and ACK after the ACK for Data Byte M. This PEC byte covers the data starting with the Slave Address through Data Byte M and is controlled by `SBRMI::Control[PECEn]`. This is the only modification to the standard SMBus PEC-optional Block Write-Block Read Process Call as defined by the SMBus Specification. The PEC byte after Data Byte N covers all previous bytes excluding the first PEC byte. Figure below shows the transmission protocol. Each byte in the protocol is sent with the most significant bit first (bit[7]). The master may reset the bus by holding the clock low for 25ms as specified by the SMBus Specification.



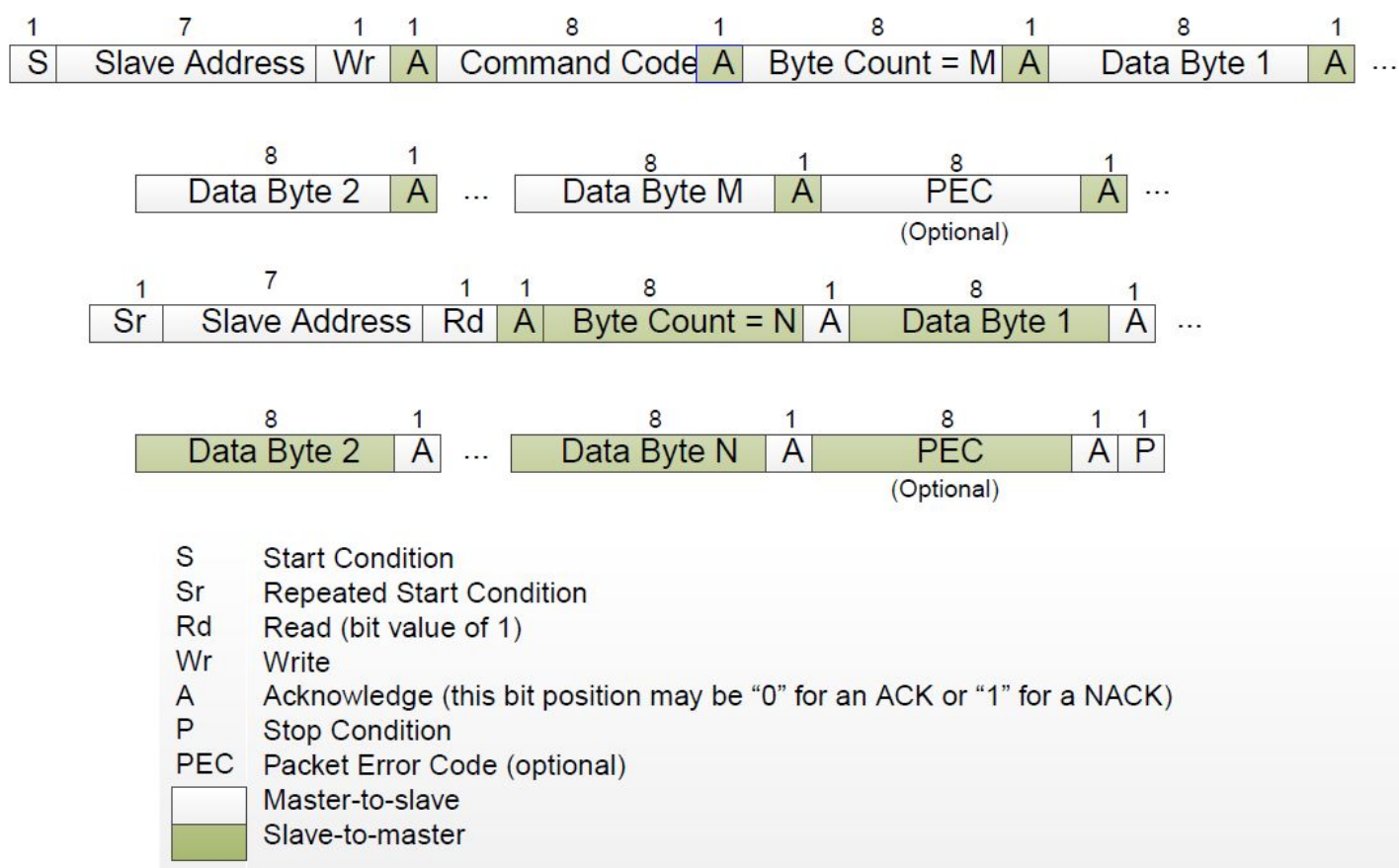


Figure 23: SBI Transmission Protocol

#### 4.4.2 SBI Remote Management Interface (SB-RMI)

SB-RMI provides an interface for an external SMBus master that can be used to perform tasks such as monitoring the processor MCA registers, processor CPUID registers etc. SB-RMI supports signaling Alert\_L when a MCE is received by any thread or when software sets SBRMI::Status[SwAlertSts]. Each package has an independent SMBUS slave port. See 4.5.1 [SBI SMBus Address].

Each package is required to contain the same number of logical threads. The SMBUS slave port attached to each package may access only the logical threads within the package. Core::X86::CpuId::SizeId identifies the number of logical threads available in a package.

##### 4.4.2.1 SB-RMI Processor State Access

The SB-RMI Functions table describes the functions for accessing processor state. See the Processor Programming Reference of the processor family for additional information about the processor registers. MSR not listed in below table is not accessible, will get "Unsupported Command" status.

Table 71: SB-RMI Functions

Function	Description	Thread Specific
CPUID	Access to CPUID registers. General purpose registers are not altered unlike a	Y



	processor CUID instruction. Use Read CUID Command Protocol where CUID Function is placed into WrData[7:4] and register is placed in WrData[8]. Access is Read-only.	
MCA Registers	Register read command using the register address to access Core::X86::Msr::MCG_CAP determines the number of MCA banks.  Use Read Processor Register Command Protocol where MSR address is placed into WrData[7:4]. Access is Read-only.	Y
DRAM Throttle	Register read or write command to access the DRAM Controller Command Throttle Register.  The thread number field is not used for this request. Writes are uniformly applied to all DRAM Controller Command Throttle Register Instances within a package. Reads return Dram Controller Command Throttle Register instance 0. Access is Read-write.	N
Mailbox Service	Soft mailbox service request to firmware for power management purposes. Past implementations allowed for mailbox operations to X86 software. No usage models for communication with x86 software exists and x86 software messaging is not supported. Access is Read-write. See mailbox specific details.	N
Boot Status	Boot Status is placed in outbound message register SBRMI::MP0OutBndMsg. Access is Read-only.	N

#### 4.4.2.1.1 SB-RMI Read Processor Register and Read CUID Commands

SB-RMI read processor register and read CUID commands are performed using the SBI Modified Block Write-Block Read Process Call. If an SMBus timeout occurs before the data is returned, a read data/status can be issued to read the data from the previous command. The previous command must be complete before a new command can be issued.

Table 72: SB-RMI Read Processor Register Command Protocol

Byte	Byte Name	Value	Notes
1	Slave Address	0111_XXX0b	Write Address.
2	Command	73h	Read CUID/Read Register Command Format.
3	WrDataLen	07h	7 Bytes.
4	WrData1	0Xh	Number of bytes to read from register. Valid values are 1 through 8.
5	WrData2	86h	Read Register command.
6	WrData3	XXXX_XXXXb	Bit[0] is Reserved. Bits[7:1] select the thread to address. 00h=Thread0. ... 7Fh=Thread127.
7	WrData4	XXh	Register Address[7:0] from the SB-RMI Functions table.
8	WrData5	XXh	Register Address[15:8] from the SB-RMI Functions table.
9	WrData6	XXh	Register Address[23:16] from the SB-RMI Functions table.
10	WrData7	XXh	Register Address[31:24] from the SB-RMI Functions table.
11	PEC	XXh	Optional PEC byte.
12	Slave Address	0111_XXX1b	Read Address.
13	RdDataLen	0Xh	Number of bytes returned = WrData1+1.

14	Status	XXh	Status Code.
15	RdData1	XXh	Register Data[7:0].
16	RdData2	XXh	Register Data[15:8]. Optional.
17	RdData3	XXh	Register Data[23:16].
18	RdData4	XXh	Register Data[31:24]. Optional.
19	RdData5	XXh	Register Data[39:32]. Optional.
20	RdData6	XXh	Register Data[47:40]. Optional.
21	RdData7	XXh	Register Data[55:48]. Optional.
22	RdData8	XXh	Register Data[63:56]. Optional.
23	PEC	XXh	Optional PEC byte.

Table 73: SB-RMI Read CPUID Command Protocol

Byte	Byte Name	Value	Notes
1	Slave Address	0111_XXX0b	Write Address.
2	Command	73h	Read CPUID/Read Register Command Format.
3	WrDataLen	08h	8 Bytes.
4	WrData1	08h	Number of CPUID bytes to read.
5	WrData2	91h	Read CPUID command.
6	WrData3	XXXX_XXXXb	Bit[0] is Reserved. Bits[7:1] select the thread to address. 00h=Thread0. ... 7Fh=Thread127.
7	WrData4	XXh	CPUID Function[7:0].
8	WrData5	XXh	CPUID Function[15:8].
9	WrData6	XXh	CPUID Function[23:16].
10	WrData7	XXh	CPUID Function[31:24].
11	WrData8	ECX[3:0]_000Xb	ECX[3:0] is the initial ECX value for extended CPUID operations. Must be 0h for non-extended operations. X: 0b=Return ebx:eax; 1b=Return edx:ecx.
12	PEC	XXh	Optional PEC byte.
13	Slave Address	0111_XXX1b	Read Address.
14	RdDataLen	09h	Number of bytes returned.
15	Status	XXh	Status Code.
16	RdData1	XXh	eax or ecx bits[7:0].
17	RdData2	XXh	eax or ecx bits[15:8].
18	RdData3	XXh	eax or ecx bits[23:16].
19	RdData4	XXh	eax or ecx bits[31:24].
20	RdData5	XXh	ebx or edx bits[7:0].
21	RdData6	XXh	ebx or edx bits[15:8].
22	RdData7	XXh	ebx or edx bits[23:16].
23	RdData8	XXh	ebx or edx bits[31:24].
24	PEC	XXh	Optional PEC byte.

Table 74: SB-RMI Read Data/Status Command Protocol

Byte	Byte Name	Value	Notes
1	Slave Address	0111_XXX0b	Write Address.
2	Command	72h	Read CPUID/Read Register Command Format.

3	WrDataLen	01h	1 byte of Write data.
4	WrData1	0Xh	Number of bytes to read from register. Valid values are 1 through 8.
5	PEC	XXh	Optional PEC byte.
6	Slave Address	0111_XXX1b	Read Address.
7	RdDataLen	0Xh	Number of bytes returned = WrData1 + 1.
8	Status	XXh	Status Code.
9	RdData1	XXh	Register Data[7:0]. Optional.
10	RdData2	XXh	Register Data[15:8]. Optional.
11	RdData3	XXh	Register Data[23:16]. Optional.
12	RdData4	XXh	Register Data[31:24]. Optional.
13	RdData5	XXh	Register Data[39:32]. Optional.
14	RdData6	XXh	Register Data[47:40]. Optional.
15	RdData7	XXh	Register Data[55:48]. Optional.
16	RdData8	XXh	Register Data[63:56]. Optional.
17	PEC	XXh	Optional PEC byte.

#### 4.4.2.1.2 SB-RMI Write Processor Register Command

Writing processor registers from SB-RMI uses two SBI Modified Block Write-Block Read Process Call commands. The first command loads the address of the register to be written into the processor. The register address loaded by this command is stored on a per-thread basis. The second command writes the data to the processor register using the stored register address. The read data/status command can be used to determine that the command completed if a SMBus timeout occurs. The previous command must be complete before a new command can be issued. WrData Address ranges beyond 32 bits are ignored.

Write Register/Load Address command is only used for DRAM throttle feature for address C001\_0079.

Table 75: SB-RMI Load Address Command Protocol

Byte	Byte Name	Value	Notes
1	Slave Address	0111_XXX0b	Write Address.
2	Command	71h	Write Register/Load Address Command Format.
3	WrDataLen	06h	6 bytes.
4	WrData1	81h	Load Address Command.
5	WrData2	XXXX_XXXXb	Bit[0] is Reserved. Bits[7:1] select the thread to address. 00h=Thread0. ... 7Fh=Thread127.
6	WrData3	XXh	Register Address[7:0] from SB-RMI Functions table.
7	WrData4	XXh	Register Address[15:8] from SB-RMI Functions table.
8	WrData5	XXh	Register Address[23:16] from SB-RMI Functions table.
9	WrData6	XXh	Register Address[31:24] from SB-RMI Functions table.
10	PEC	XXh	Optional PEC byte.
11	Slave Address	0111_XXX1b	Read Address.
12	RdDataLen	01h	Number of bytes returned.
13	Status	XXh	Status Code.
14	PEC	XXh	Optional PEC byte.

Table 76: SB-RMI Write Processor Register Command Protocol

Byte	Byte Name	Value	Notes
1	Slave Address	0111_XXX0b	Write Address.
2	Command	71h	Write Register/Load Address Command Format.
3	WrDataLen	0Xh	Total number of WrData bytes sent by the master. The total number of bytes written to the register (WrDataLen - 2) must match the size of the register that is being written or undefined data will be written into the register.
4	WrData1	87h	Write Register Command.
5	WrData2	XXXX_XXXXb	Bit[0] is Reserved. Bits[7:1] select the thread to address. 00h=Thread0. ... 7Fh=Thread127.
6	WrData3	XXh	Register Data[7:0].
7	WrData4	XXh	Register Data[15:8]. Optional.
8	WrData5	XXh	Register Data[23:16]. Optional.
9	WrData6	XXh	Register Data[31:24]. Optional.
10	WrData7	XXh	Register Data[39:32]. Optional.
11	WrData8	XXh	Register Data[47:40]. Optional.
12	WrData9	XXh	Register Data[55:48]. Optional.
13	WrData10	XXh	Register Data[63:56]. Optional.
14	PEC	XXh	Optional PEC byte.
7+WrDataLen	Slave Address	0111_XXX1b	Read Address.
8+WrDataLen	RdDataLen	01h	Number of bytes returned.
9+WrDataLen	Status	XXh	Status Code.
10+WrDataLen	PEC	XXh	Optional PEC byte.

#### 4.4.2.1.3 SB-RMI Protocol Status Codes

The legal values for the Status byte of the SB-RMI processor state accesses are shown in the following table.

Table 77: SB-RMI Status Codes

Status Code	Name	Description
00h	Success	Command.
11h	Command Timeout	Command did not complete before an SMBus timeout occurred. This status code will never occur if (SBRMI_x01[TimeoutDis] == 1). MP has not sent the request to CPU/NB.
22h	Warm reset	A warm reset occurred during the transaction.
40h	Unknown Command Format	The value in Command Format field is not recognized.
41h	Invalid Read Length	The value in RdDataLen is less than 1 or greater than 32.
42h	Excessive Data Length	The sum of the RdDataLen and WrDataLen is greater than 32 and RdDataLen is greater than or equal to 1 and less than or equal to 32.
44h	Invalid thread	Invalid thread selected.
45h	Unsupported	Command not supported by the processor.

	Command	
81h	Command Aborted	The processor core targeted by the command could not start the command and was aborted by the processor.

#### 4.4.2.2 SB-RMI Mailbox Service

SB-RMI supports soft mailbox service request to MP1 (power management firmware) through SBRMI inbound/outbound message registers. The message type is defined in the following table.

Table 78: SB-RMI Soft Mailbox Message

Command	Message	Description	Command Data In	Command Data Out
01h..2Fh	Reserved	N/A	N/A	N/A
30h	WRITE_SUSTAINED_POWER_LIMIT	Set Sustained power limit for SOC package.	[31:0]=SPL in mW.	None
31h	WRITE_FAST_PT_LIMIT	Set APU power limit for system power supply peak control.	[31:0]=fPPT in mW.	None
32h	WRITE_SLOW_PPT_LIMIT	Set APU power limit for system power supply thermal control.	[31:0]=sPPT in mW.	None
33h	WRITE_SLOW_PPT_TIME_CONSTANT	Set residency at fPPT.	[31:0]=sPPt Time Constant in seconds.	None
34h	WRITE_THERM_CTL_LIMIT	Set the thermal throttling limit.	[31:0]=Therm limit in degree Celsius.	None
35h	WRITE_VRM_VDD_CURRENT_LIMIT	Set VDDCR_VDD TDC.	[31:0]=VDD TDC in mA.	None
36h	WRITE_VRM_VDD_MAXIMUM_CURRENT_LIMIT	Set VDDCR_VDD EDC.	[31:0]=VDD EDC in mA.	None
37h	WRITE_VRM_SOC_CURRENT_LIMIT	Set VDDCR_SOC TDC.	[31:0]=SOC TDC in mA.	None
38h	WRITE_VRM_SOC_MAXIMUM_CURRENT_LIMIT	Set VDDCR_SOC EDC.	[31:0]=SOC EDC in mA.	None
39h	WRITE_PROCHOT_L_DEASSERTION_RAMP_TIME	Set the PROCHOT_L deassertion ramp time to take to ramp the CCLK/GFXCLK clocks up to Fmax from Fmin when PROCHOT_L is deasserted. A value of zero means to use the default (20ms) value.	[31:0]=Steps.	None
3Ah	WRITE_STT_SENSOR_VALUE	Used to send the PCB sensor temperature data for System temperature tracking.	32-bits. [31:24]=Unused. [23:16]=Sensor index.	None

			[15:0]=Temperature as signed integer with 8 fractional bits.	
3Bh	WRITE_PeAPM_SLOW_PPT_LIMIT_APU	Set the maximum sPPT value APU is allowed to consume when smartshift is enabled.	[31:0]=sPPT (APU only) in mW.	None

#### 4.4.2.2.1 SB-RMI Mailbox Sequence

The sequence is as follows:

1. The initiator (BMC) indicates that command is to be serviced by firmware by writing 80h to SBRMI::InBndMsg\_inst7 (SBRMI\_x3F). This register must be set to 80h after reset.
2. The initiator (BMC) writes the command to SBRMI::InBndMsg\_inst0 (SBRMI\_x38).
3. For Write operations or Read operations, which require additional addressing information as shown in Table 78 [SB-RMI Soft Mailbox Message] above, the initiator (BMC) writes Command Data In[31:0] to SBRMI::InBndMsg\_inst[4:1] {SBRMI\_x3C(MSB):SBRMI\_x39(LSB)}.
4. The initiator (BMC) writes 01h to SBRMI::SoftwareInterrupt to notify firmware to perform the requested Read or Write command.
5. Firmware reads the message and performs the defined action.
6. Firmware writes the original command to outbound message register SBRMI::OutBndMsg\_inst0 (SBRMI\_x30).
7. Firmware writes SBRMI::Status[SwAlertSts] = 1 to generate an ALERT (if enabled) to initiator (BMC) to indicate completion of the requested command. Firmware must (if applicable) put the message data into the message registers SBRMI::OutBndMsg\_inst[4:1] {SBRMI\_x34(MSB):SBRMI\_x31(LSB)}.
8. Firmware clears the interrupt on SBRMI::SoftwareInterrupt.
9. For a Read operation, the initiator (BMC) reads the firmware response Command Data Out[31:0] from SBRMI::OutBndMsg\_inst[4:1] {SBRMI\_x34(MSB):SBRMI\_x31(LSB)}.
10. BMC must write 1'b1 to SBRMI::Status[SwAlertSts] to clear the ALERT to initiator (BMC). It is recommended to clear the ALERT upon completion of the current mailbox command.

Table 79: SB-RMI Soft Mailbox Error Code

Error Type	Description	Code
No error	Mailbox message command executed successfully without an error.	00h
Command Aborted	Mailbox message command was aborted due to internal error. DataOut must be ignored.	01h
Unknown Command	Unknown mailbox message.	02h
Invalid Core	Invalid core is specified in mailbox message parameters.	03h

The mailbox error code is written by Firmware in SBRMI::OutBndMsg\_inst7 (SBRMI\_x37).

#### 4.4.2.3 SB-RMI Boot code status

Boot code will dump the dynamic boot status into SBRMI::MP0OutBndMsg. BMC can then just read this status through SBI interface to determine progress through the boot flow.

#### 4.4.2.4 SB-RMI Register Access

The SB-RMI registers can be read or written from the SMBus interface using the SMBus defined PEC-optional Read Byte and Write Byte protocols with the SB-RMI register number in the command byte or the PEC-optional Block Read and Block Write protocols with the first SB-RMI register number to be accessed in the command byte. Block Read/Write

protocol access for SB-RMI registers is controlled by SBRMI::Control[BlkRWEn]. The SB-RMI interface supports Block Writes of up to 32 bytes, and Block Reads of up to 32 bytes as specified by SBRMI::ReadSize[RdSize]. Bytes are returned in ascending register order starting with the first SB-RMI register in the command byte.

#### 4.4.2.4.1 SB-RMI Register Block Access

The following example shows a write from SBRMI\_x18 to SBRMI\_x1F using SMBus Block Write protocol with SBRMI::Control[BlkRWEn] set to 1.

*Table 80: SB-RMI Register Block Write Protocol*

Byte	Byte Name	Value	Notes
1	Slave Address	0111_XXX0b	Write Address.
2	Command	18h	Indicates starting register SBRMI_x18.
3	Byte Count	08h	Number of bytes to write.
4	Data Byte 1	00h	Write a value to SBRMI_x18h.
5	Data Byte 2	00h	Write a value to SBRMI_x19h.
6	Data Byte 3	00h	Write a value to SBRMI_x1Ah.
7	Data Byte 4	00h	Write a value to SBRMI_x1Bh.
8	Data Byte 5	00h	Write a value to SBRMI_x1Ch.
9	Data Byte 6	00h	Write a value to SBRMI_x1Dh.
10	Data Byte 7	00h	Write a value to SBRMI_x1Eh.
11	Data Byte 8	00h	Write a value to SBRMI_x1Fh.
12	PEC	XXh	Optional PEC byte.

The following example shows a read from SBRMI\_x10 to SBRMI\_x17 using SMBus Block Read protocol with SBRMI::Control[BlkRWEn] set to 1 and SBRMI::ReadSize[RdSize] set to 8.

*Table 81: SB-RMI Register Block Read Protocol*

Byte	Byte Name	Value	Notes
1	Slave Address	0111_XXX0b	Write Address.
2	Command	10h	Indicates starting register SBRMI_x10.
3	Slave Address	0111_XXX1b	Read Address.
4	Byte Count	08h	Number of bytes to read.
5	Data Byte 1	00h	Read a value from SBRMI_x10h.
6	Data Byte 2	00h	Read a value from SBRMI_x11h.
7	Data Byte 3	00h	Read a value from SBRMI_x12h.
8	Data Byte 4	00h	Read a value from SBRMI_x13h.
9	Data Byte 5	00h	Read a value from SBRMI_x14h.
10	Data Byte 6	00h	Read a value from SBRMI_x15h.
11	Data Byte 7	00h	Read a value from SBRMI_x16h.
12	Data Byte 8	00h	Read a value from SBRMI_x17h.
13	PEC	XXh	Optional PEC byte.

#### 4.4.2.4.2 SB-RMI Register Byte Access

The following example shows a write to SBRMI\_x03 using the SMBus Write Byte protocol with SBRMI::Control[BlkRWEn] set to 0.



*Table 82: SB-RMI Register Write Byte Protocol*

Byte	Byte Name	Value	Notes
1	Slave Address	0111_XXX0b	Write Address.
2	Command	03h	Indicates SB-RMI register 03.
3	Data Byte	04h	Write a value of 04h.
4	PEC	XXh	Optional PEC byte.

The following example shows a read from SBRMI\_x03 using the SMBus Read Byte protocol with SBRMI::Control[BlkRWEn] set to 0.

*Table 83: SB-RMI Register Read Byte Protocol*

Byte	Byte Name	Value	Notes
1	Slave Address	0111_XXX0b	Write Address.
2	Command	03h	Indicates SB-RMI register 03.
3	Slave Address	0111_XXX1b	Read address.
4	Data Byte	04h	Value of SBRMI_x03h.
5	PEC	XXh	Optional PEC byte.

#### 4.4.2.5 SB-RMI Alert

The processor alerts the SBI when a Machine Check Exception occurs within the system. The Machine Check Exception status is reflected in registers SBRMI\_x01[F:0].

The processor alerts the SBI on system fatal error event. This status is reflected in SBRMI\_x02[SwAlertSts]. To enable this functionality, SBRMI\_x01[SwAlertMask] must be clear.

### 4.4.3 SBI Error Detection and Recovery

This section describes the various error detection and recovery methods that can be used on the SBI bus. The important item in providing a high reliability SBI connection is the ability to detect when an error occurs and to gracefully recover from that error. When the SBI connections are noisy, messages can become garbled which, in turn, may cause undefined behavior on the SBI bus. The most common noise sources are cross-talk and clock skew. Cross-talk results when the SBI connections are routed too close to other signal carrying lines. Clock skew is usually a result of higher than expected capacitance, between the SBI signals (clock and / or data) and ground, which causes the master and slave devices to disagree on when data should be stable and when it is allowed to be changing.

#### 4.4.3.1 Error Detection

SBI provides several methods of error detection: protocol ACK/NAK, packet error correction (PEC) fields, and timeouts. The ACK/NAK mechanism is always active in SBI, but the PEC and timeouts are optional.

##### 4.4.3.1.1 ACK/NAK Mechanism

After each byte of an SBI message, the device receiving that byte must either acknowledge (ACK) that it received the byte correctly, or deny (NAK) that the byte was correctly received. This is most easily seen in the case of the address bytes which follow a START (or REPEATED START) sequence, but can be used anywhere in the message. In the case of an address byte, if a slave device recognizes the address, it will respond with an ACK and await the rest of the message. If a slave device does not recognize the message, it will respond with a NAK and ignore the rest of the message.



#### 4.4.3.1.2 Packet Error Correction (PEC)

The RMI protocols allow for PEC bytes to be appended to messages. The sending side calculates the PEC, based on the data it intends to transmit, and appends it to the transmitted data. The receiving side calculates the PEC based on the data it actually receives and compares that to the PEC it receives. If the two PECs do not match, an error has occurred and the message should be discarded. When a device detects a PEC mismatch, it should send a NAK in response to the PEC. No special programming is needed to enable the PEC on AMD devices. If the PEC is present on an incoming message, the device will verify the PEC and ACK or NAK as appropriate. The PEC is always calculated on outgoing messages. It is up to the bus master to request the PEC by sending clocks for that byte before sending either a NAK or a STOP sequence.

#### 4.4.3.1.3 Bus Timeouts

Bus timeouts should be enabled to prevent a device waiting indefinitely on a message that may not be coming. Some timeouts are used to prevent the SBI bus from waiting for a response from a CPU that is in a power-saving idle mode. Other timeouts are used to allow the slave device to recognize that the bus master is attempting to reset all of the devices on the SBI bus. Either way, when a device recognizes a timeout, it should abort its current message transfer.

#### 4.4.3.2 Error Recovery

The simplest form of error recovery is a retry. When the bus master detects an unexpected NAK, it should abort the current transfer and retry the message sequence. In some cases, however, a message can be so garbled that a simple retry is insufficient. This can occur, if there are multiple devices on the bus and a garbled address byte has caused the wrong slave device to be selected. That slave device may even continue to transmit during the retry. In those cases, it will be necessary to force a reset of all devices on the SBI bus, before retrying the message transfer.

##### 4.4.3.2.1 SBI Bus Reset

The bus master can hold the clock low for a period longer than the standard timeout in order to force slave devices off the bus (see docSMB section 3.1.1.3 of the System Management Bus (SMBus) Specification, version 2.0). All SBI slave devices are required to reset their communications if another device holds the clock line low for longer than TTimeout, min (25 milliseconds). The devices are required to complete their reset within TTimeout, max (35 milliseconds). SBI bus masters should use the extended timeout to force a reset of all slave devices if a simple retry does not remove an error condition.

### 4.5 SBI Physical Interface

#### 4.5.1 SBI SMBus Address

The SMBus address is really 7 bits. Some vendors and the SMBus specification show the address as 8 bits: bits[7:1] as the left-justified address, and bit[0] as the Read/Write flag, where 0 indicates a Write and 1 indicates a Read. Some vendors use only the 7 bits to describe the address.

#### 4.5.2 SBI Bus Timing

SBI supports 100KHz standard-mode and 400 KHz fast-mode I2C operation. Refer to the standard-mode and fast-mode timing parameters in the I2C specification.

### 4.6 SB-RMI Registers

Reads to unimplemented registers may return non zero value. Writes to unimplemented registers are discarded.

**SBRMIx00 [Revision] (SBRMI::Revision)**

Read-only. Reset: 10h.

Bits	Description
7:0	<b>Revision: SB-RMI revision.</b> Read-only. Reset: 10h. This field specifies the APML specification revision that the product is compliant to. 0x10=1.0x Revision.

**SBRMIx01 [Control] (SBRMI::Control)**

Read-write. Reset: 01h.

Bits	Description
7	<b>PECEn: packet error checking enable.</b> Read-write. Reset: 0. This only controls the intermediate PEC of the SBI Modified Block Write-Block Read Process Call. 0=Intermediate PEC is disabled. 1=Intermediate PEC is enabled.
6:5	Reserved.
4	<b>SwAlertMask: software alert mask.</b> Read-write. Reset: 0. 0=Alert_L signaling is enabled when SBRMI_x02SwAlertSts is set. 1=Alert_L signaling is disabled when SBRMI_x02SwAlertSts is set.
3	<b>BlkRWEn: block read/write enable.</b> Read-write. Reset: 0. Controls Block Read/Write access to register ranges SBRMI_x[4F:10] and SBRMI_x[9F:80]. 0=SMBus accesses can only use the Byte Read/Write protocol. 1=SMBus accesses can only use the Block Read/Write protocol. NOTE: All other register ranges only support Byte Read/Write access, independent of the state of the BlkRWEn control bit.
2	<b>TimeoutDis: SB-RMI timeout disable.</b> Read-write. Reset: 0. 1=SMBus defined timeouts are disabled. If the SB-TSI interface is also in use, SMBus timeouts should be enabled or disabled in a consistent manner on both interfaces. The SB-TSI timeout setting is used by SB-RMI until the SMBus interface can determine which interface is targeted by the transaction.
1	<b>AraDis: SB-RMI ARA disable.</b> Read-write. Reset: 0. 1=Sending of an ARA response is disabled. 0=Sending of an ARA response is enabled.
0	<b>AlertMask: SB-RMI alert mask.</b> Read-write. Reset: 1. Read-write; set-by-hardware if AraDis=0 and a successful ARA is sent. 1=Alert_L signaling disabled. 0=Alert_L is asserted if any unmasked event is present in the [The Alert Status Registers] SBRMI_x1[F:0], or if SBRMI_x02[SwAlertSts] == 1 and SwAlertMask == 0.

**SBRMIx02 [Status] (SBRMI::Status)**

Reset: 00h.

Bits	Description
7:2	Reserved.
1	<b>SwAlertSts: SB-RMI software alert status.</b> Read-write, Volatile. Reset: 0. Write-one-to-clear from the SMBus interface; Read-write from the processor. Set by firmware as a result of a Machine Check Exception prior to the MCE related warm reset. Set by firmware to indicate the completion of a mailbox operation.
0	<b>AlertSts: SB-RMI alert status.</b> Read-only, Volatile. Reset: 0. Read-only. 1=Alert event present in SBRMI::AlertStatus.

**SBRMIx03 [Read Size] (SBRMI::ReadSize)**

Read-write. Reset: 01h.

This register specifies the number of bytes to return when using the block read protocol to read SBRMI\_x[4F:10] and SBRMI\_x[90:80].

Bits	Description
7:6	Reserved.
5:0	<b>RdSize: read size.</b> Read-write. Reset: 01h. Specifies the number of bytes to return when using the block read protocol.
<b>ValidValues:</b>	
<b>Value</b>	<b>Description</b>
00h	Reserved.
20h-01h	<Value> bytes.
3Fh-21h	Reserved.

**SBRMIx0[4...5] [Thread Enable Status] (SBRMI::ThreadEnableStatus)**

Read-only.

\_inst[1:0]; SBRMIx0[5:4]

**Bits Description****7:0 threadEnStat: thread enable status.** Read-only.**Description:** 1=Thread is enabled.

Offset[7:0]	inst	Description
04h	0	Threads[7:0].
05h	1	Threads[15:8].

**SBRMIx1[0...F] [Alert Status] (SBRMI::AlertStatus)**

Read, Write-1-to-clear, Volatile.

\_inst[15:0]; SBRMIx1[F:0]

**Bits Description****7:4** Reserved.**3:0 MceStat: MCE status.** Read, Write-1-to-clear, Volatile.**Description:** Bit vector for threads. 1=MCE occurred for thread. Set by hardware.

Offset[7:0]	inst	Description
10h	0	Threads[48,32,16,0].
11h	1	Threads[49,33,17,1].
12h	2	Threads[50,34,18,2].
13h	3	Threads[51,35,19,3].
14h	4	Threads[52,36,20,4].
15h	5	Threads[53,37,21,5].
16h	6	Threads[54,38,22,6].
17h	7	Threads[55,39,23,7].
18h	8	Threads[56,40,24,8].
19h	9	Threads[57,41,25,9].
1Ah	10	Threads[58,42,26,10].
1Bh	11	Threads[59,43,27,11].
1Ch	12	Threads[60,44,28,12].
1Dh	13	Threads[61,45,29,13].
1Eh	14	Threads[62,46,30,14].
1Fh	15	Threads[63,47,31,15].

**SBRMIx2[0...F] [Alert Mask] (SBRMI::AlertMask)**

Read-write.

\_inst[15:0]; SBRMIx2[F:0]

Bits	Description	
7:4	Reserved.	
3:0	<b>MceAlertMsk: MCE alert mask.</b> Read-write. <b>Description:</b> Bit vector for threads. 1=Alert signaling disabled for corresponding SBRMI::AlertStatus[MceStat] for thread.	
	Offset[7:0]	inst      Description
	20h	0      Threads[48,32,16,0].
	21h	1      Threads[49,33,17,1].
	22h	2      Threads[50,34,18,2].
	23h	3      Threads[51,35,19,3].
	24h	4      Threads[52,36,20,4].
	25h	5      Threads[53,37,21,5].
	26h	6      Threads[54,38,22,6].
	27h	7      Threads[55,39,23,7].
	28h	8      Threads[56,40,24,8].
	29h	9      Threads[57,41,25,9].
	2Ah	10     Threads[58,42,26,10].
	2Bh	11     Threads[59,43,27,11].
	2Ch	12     Threads[60,44,28,12].
	2Dh	13     Threads[61,45,29,13].
	2Eh	14     Threads[62,46,30,14].
	2Fh	15     Threads[63,47,31,15].

**SBRMIx3[0...7] [Out-Bound Message] (SBRMI::OutBndMsg)**

Read-write. Reset: 00h.

\_inst[7:0]; SBRMIx3[7:0]

Bits	Description
7:0	<b>OutBndMsg: outbound message data.</b> Read-write. Reset: 00h.
	<b>Description:</b> Read-write from the processor; Read-only from the SMBus interface.
	Usage convention is:
	<ul style="list-style-type: none"><li>SBRMI::OutBndMsg_inst0 is command copied by firmware from SBRMI::InBndMsg_inst0.</li><li>SBRMI::OutBndMsg_inst[4:1] are 32-bit data.</li><li>SBRMI::OutBndMsg_inst[6:5] are Reserved.</li><li>SBRMI::OutBndMsg_inst[7] contains Mailbox Error Code, per Table 79 [SB-RMI Soft Mailbox Error Code]</li></ul>

**SBRMIx3[8...F] [In-Bound Message] (SBRMI::InBndMsg)**

Read-write. Reset: 00h.

\_inst[7:0]; SBRMIx3[F:8]

Bits	Description																											
7:0	<b>InBndMsg: inbound message data.</b> Read-write. Reset: 00h.																											
	<b>Description:</b> Read-write from the SMBus interface; Read-only from the processor. These registers are used for communicating 32-bit messages from BMC to firmware.																											
	Usage convention is:																											
	<ul style="list-style-type: none"><li>• SBRMI::InBndMsg_inst0 is command.</li><li>• SBRMI::InBndMsg_inst[4:1] are 32-bit data.</li><li>• SBRMI::InBndMsg_inst[6:5] are Reserved.</li><li>• SBRMI::InBndMsg_inst7: Bit[7] Must be 1'b1 to send message to firmware.</li></ul>																											
	<table><tr><th>Offset[7:0]</th><th>inst</th><th>Description</th></tr><tr><td>38h</td><td>0</td><td>Inbound message 0.</td></tr><tr><td>39h</td><td>1</td><td>Inbound message 1.</td></tr><tr><td>3Ah</td><td>2</td><td>Inbound message 2.</td></tr><tr><td>3Bh</td><td>3</td><td>Inbound message 3.</td></tr><tr><td>3Ch</td><td>4</td><td>Inbound message 4.</td></tr><tr><td>3Dh</td><td>5</td><td>Inbound message 5.</td></tr><tr><td>3Eh</td><td>6</td><td>Inbound message 6.</td></tr><tr><td>3Fh</td><td>7</td><td>Inbound message 7.</td></tr></table>	Offset[7:0]	inst	Description	38h	0	Inbound message 0.	39h	1	Inbound message 1.	3Ah	2	Inbound message 2.	3Bh	3	Inbound message 3.	3Ch	4	Inbound message 4.	3Dh	5	Inbound message 5.	3Eh	6	Inbound message 6.	3Fh	7	Inbound message 7.
	Offset[7:0]	inst	Description																									
	38h	0	Inbound message 0.																									
	39h	1	Inbound message 1.																									
	3Ah	2	Inbound message 2.																									
	3Bh	3	Inbound message 3.																									
3Ch	4	Inbound message 4.																										
3Dh	5	Inbound message 5.																										
3Eh	6	Inbound message 6.																										
3Fh	7	Inbound message 7.																										

**SBRMIx40 [Software Interrupt] (SBRMI::SoftwareInterrupt)**

Read,Write-1-only. Reset: 00h.

This register is used by the SMBus master to generate an interrupt to the processor to indicate that a message is available.

Bits	Description
7:1	Reserved.
0	<b>SwInt: firmware interrupt.</b> Read,Write-1-only. Reset: 0. Read,Write-1-only from the SMBus interface; Read,Write-1-to-clear from firmware. 1=Indicates a firmware mailbox service request.

**SBRMIx41 [Thread Number] (SBRMI::ThreadNumber)**

Read-write. Reset: 00h.

This register indicates the maximum number of threads present.

Bits	Description
7:0	<b>threadNum: thread number.</b> Read-write. Reset: 00h. Read-only from the SMBus interface. Specifies the maximum number of threads present. Format is [6:1] number of threads – 40h and range of available threads 40h – 01h. Firmware loads the initial value based on the maximum number of threads available after any fused off or soft-down-coring is complete.

**SBRMIx8[0...7] [MP0 Out-Bound Message] (SBRMI::MP0OutBndMsg)**

Read-write. Reset: 00h.

\_inst[7:0]; SBRMIx8[7:0]

Bits	Description	
7:0	<b>MP0OutBndMsg: outbound message data.</b> Read-write. Reset: 00h. <b>Description:</b> Read-write from the processor; Read-only from the SMBus interface. These registers are used for sending messages from PSP firmware running on the MP0 to the SMBus master. MP0 boot status is dynamically written to this register during the boot process.	
	Offset[7:0]	Description
	80h	MP0 Outbound message 0.
	81h	MP0 Outbound message 1.
	82h	MP0 Outbound message 2.
	83h	MP0 Outbound message 3.
	84h	MP0 Outbound message 4.
	85h	MP0 Outbound message 5.
	86h	MP0 Outbound message 6.
	87h	MP0 Outbound message 7.

## 5 SB Temperature Sensor Interface (SB-TSI)

### 5.1 Overview

The SBI temperature sensor interface (SB-TSI) is an emulation of the software and physical interface of a typical 8-pin remote temperature sensor (RTS), see Figure 24 [RTS Thermal Management Example]. The goal is to resemble a typical RTS so that KBC or BMC firmware requires minimal changes for future AMD products, see Figure 25 [SB-TSI Thermal Management Example]. SB-TSI supports the SMBus protocols that typical RTS supports.

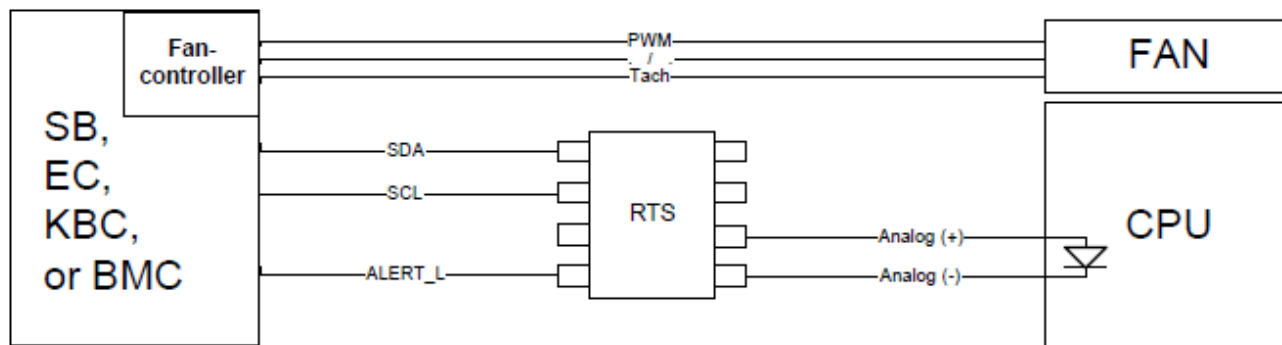


Figure 24: RTS Thermal Management Example

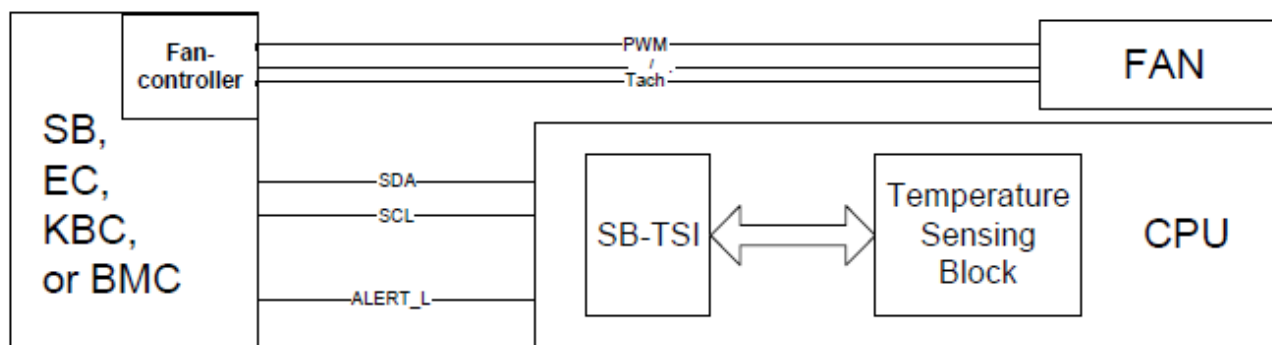


Figure 25: SB-TSI Thermal Management Example

Refer to the following external sources for additional information.

- System Management Bus (SMBus) specification. See docSMB.
- I2C-bus Specification and User Manual, Revision 03. See docI2C.

### 5.1.1 Definitions

Table 84: SB-TSI Definitions

Term	Description
<b>BMC</b>	Base management controller.
<b>TCC</b>	Temperature calculation circuit.
<b>Tctl</b>	Processor temperature control value.
<b>TSM</b>	Temperature sensor macro.
<b>SB-TSI</b>	Sideband Internal Temperature Sensor Interface. See APML.

## 5.2 SB-TSI Protocol

The SB-TSI largely follows SMBus v2.0 specification except:

- The combined-format repeated start sequence is not supported in standard-mode and fast-mode. The response of the processor's SB-TSI to the sequence is undefined.
- Only 7-bit SMBus addresses are supported.
- SB-TSI implements the Send/Receive Byte and Read/Write Byte protocols.
- SB-TSI registers can only be written by using a Write byte command.
- Address Resolution Protocol (ARP) is not supported.
- Packet Error Checking (PEC) is not supported.
- The usage of unsupported protocols may lead to an undefined bus condition.
- To release the bus from an undefined condition and to reset the SB-TSI slave, the bus master must hold the clock low for a duration of time that is longer than Ttimeout.max, as specified for SMBus. The time-out needs to be enabled by SBTsi::TimeoutConfig[TimeoutEn] = 1.

### 5.2.1 SB-TSI Send/Receive Byte Protocol

A SMBus master can Read SB-TSI registers by issuing a send byte command with the address of the register to be read as the data byte followed by a receive byte command.

#### 5.2.1.1 SB-TSI Address Pointer

The SB-TSI controller has an internal address pointer that is updated when a register is accessed using a Read or Write byte command or when a send byte command is received. This address pointer is used to determine the address of the register being read when a receive byte command is processed by the controller.

### 5.2.2 SB-TSI Read/Write Byte Protocol

An SMBus master can Read or Write SB-TSI registers by issuing a Read or a Write byte command with the address of the register to be read or written in the command code field.

### 5.2.3 Alert Behavior

The ALERT\_L pin is asserted if (SBTSI::Status[TempHighAlert] || SBTsi::Status[TempLowAlert]) && ~SBTSI::Config[AlertMask] as shown in Figure 3. The following registers also affect temperature alert behavior.

- SBTsi::Config[AraDis]: Disables ARA response.
- SBTsi::UpdateRate[UpRate]: Specifies rate at which temperature thresholds are checked.
- {SBTSI::HiTempInt[HiTempInt], SBTsi::HiTempDec[HiTempDec]}: Sets high temperature threshold.
- {SBTSI::LoTempInt[LoTempInt], SBTsi::LoTempDec[LoTempDec]}: Sets low temperature threshold.



- SBTSI::AlertThreshold[AlertThr]: Specifies number of consecutive temperature samples to assert an alert.
- SBTSI::AlertConfig[AlertCompEn]: Specifies ALERT\_L pin to be in latched or comparator mode. Affects ARA.

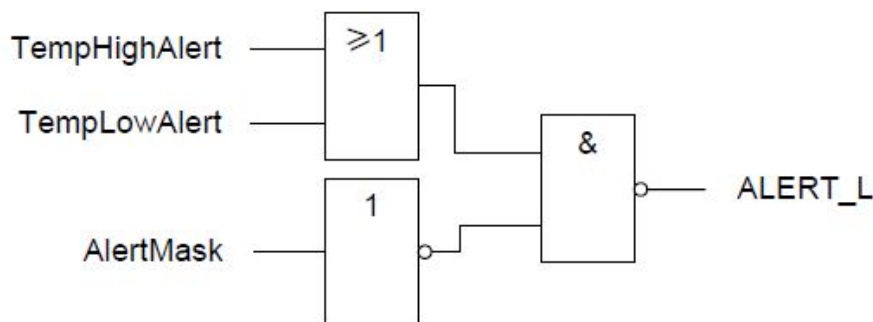


Figure 26: Alert Assertion Diagram

#### 5.2.4 Atomic Read Mechanism

To ensure that the two required Reads (integer and decimal) for reading the CPU temperature are always originated from one temperature value, atomic reading procedures are required. SB-TSI offers functions to maintain atomicity between the temperature integer and decimal bytes.

SBTSI::Config[ReadOrder] specifies the order for reading below registers:

- Integer and decimal part of SBTSI::CpuTempInt and SBTSI::CpuTempDec

If SBTSI::Config[ReadOrder] is 0, then a read of the integer part triggers a latch of the decimal part until the next read of the integer part. This latch syncs the decimal part with the integer part. The integer part is continuously updated.

If SBTSI::Config[ReadOrder] is 1, then the Read order to ensure atomicity is Reversed, i.e., decimal part = first, integer part = second.

If it is not possible to ensure a dedicated Read order as described above, the Run/Stop bit ([The SB-TSI Configuration Register] SBTSI::Config[RunStop]) may be used to provide atomicity of reading the CPU temperature. If this bit is 0, the CPU temperature registers are updated continuously. If it is 1, they get frozen and always deliver their last value on Read requests.

- Set SBTSI::Config[RunStop].
- Read the integer (SBTSI::CpuTempInt) or the decimal (SBTSI::CpuTempDec) part of the CPU temperature.
- Read the remaining part of the CPU temperature.
- Clear SBTSI::Config[RunStop].

#### 5.2.5 SB-TSI Temperature and Threshold Encodings

SB-TSI CPU temperature readings and limit registers encode the temperature in increments of 0.125 from 0 to 255.875. The high byte represents the integer portion of the temperature from 0 to 255. One increment in the high byte is equivalent to a step of one. The upper three bits of the low byte represent the decimal portion of the temperature. One increment of these bits is equivalent to a step of 0.125.

*Table 85: SB-TSI CPU Temperature and Threshold Encoding Examples*

Temperature	Temperature High Byte SBTSI::CpuTempInt[CpuTempInt] SBTSI::HiTempInt[HiTempInt] SBTSI::LoTempInt[LoTempInt]	Temperature Low Byte SBTSI::CpuTempDec[CpuTempDec] SBTSI::HiTempDec[HiTempDec] SBTSI::LoTempDec[LoTempDec]
0.000 °C	0000_0000b	0000_0000b
1.000 °C	0000_0001b	0000_0000b
25.125 °C	0001_1001b	0010_0000b
50.875 °C	0011_0010b	1110_0000b
90.000 °C	0101_1010b	0000_0000b

## 5.2.6 SB-TSI Temperature Offset Encoding

By default, SBTISI::CpuTempInt and SBTISI::CpuTempDec provide Tctl from the processor. The temperature offset registers allow the system to adjust the SB-TSI temperature from Tctl.

The SB-TSI temperature offset registers use a different encoding in order to provide negative temperature values. SBTISI::CpuTempOffInt[CpuTempOffInt] and SBTISI::CpuTempOffDec[CpuTempOffDec] form an 11-bit, 2's complement value representing the temperature offset. The high byte encodes the integer portion of the temperature and the upper three bits of the low byte represent the fractional portion of the temperature offset. One increment of these bits is equivalent to a step of 0.125 °C. After reset the offset is always set to 0 °C. Software needs to adjust the offset to the appropriate level.

*Table 86: SB-TSI Temperature Offset Encoding Examples*

Temperature	Temperature High Byte SBTSI::CpuTempOffInt[CpuTempOffInt]	Temperature Low Byte SBTSI::CpuTempOffDec[CpuTempOffDec]
-10.375 °C	1111_0101b	1010_0000b
-0.250 °C	1111_1111b	1100_0000b
0.000 °C	0000_0000b	0000_0000b
0.875 °C	0000_0000b	1110_0000b
10.000 °C	0000_1010b	0000_0000b

## 5.3 SB-TSI Physical Interface

This chapter describes the physical interface of the SB-TSI.

### 5.3.1 SB-TSI SMBus Address

The SMBus address is really 7 bits. Some vendors and the SMBus specification show the address as 8 bits: bits[7:1] as the left-justified address, and bit[0] as the Read/Write flag, where 0 indicates a Write and 1 indicates a Read. Some vendors use only the 7 bits to describe the address. The addresses can vary with address select pins.

*Table 87: SB-TSI Address Encodings*

Socket ID	SB-TSI Address
0b	98h for 8-bit or 4Ch for 7-bit.
1b	90h for 8-bit or 48h for 7-bit.

### 5.3.2 SB-TSI Bus Timing

SB-TSI supports standard-mode (100 kHz) and fast-mode (400 kHz) according to the I2C-bus Specification and User Manual.

### 5.3.3 SB-TSI Bus Electrical Parameters

SB-TSI conforms to most of the I2C fast-mode electrical parameters. See the Electrical Data Sheet for the processor family for electrical parameters.

### 5.3.4 Pass-FET Option

The KBC may not have the capability to directly interface to SB-TSI. Pass FETs may be used to create two SMBus segments, as shown in the following diagram.

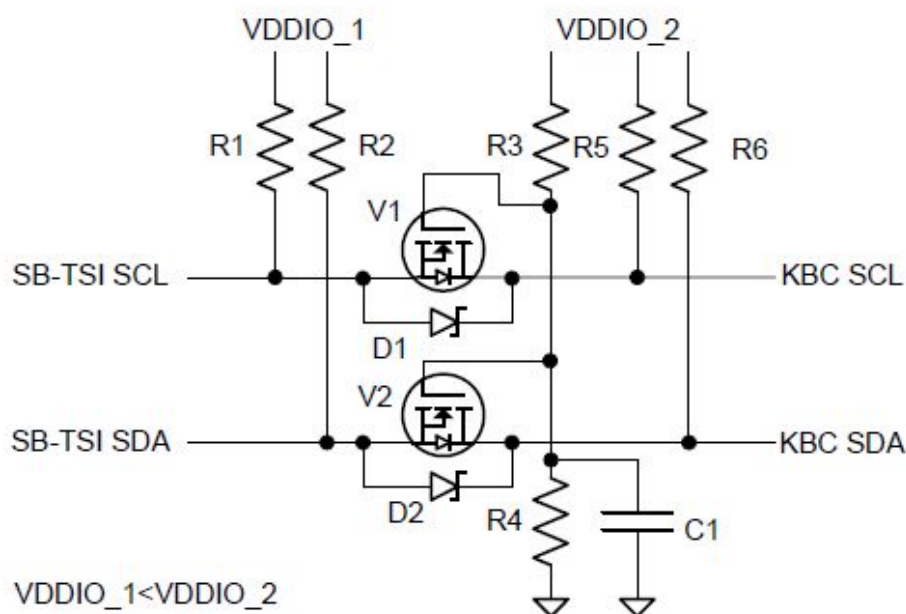


Figure 27: Pass FET Implementation

#### Notes:

- SCL and SDA pull-up resistors (R5 and R6, respectively) are the normal pull-up resistors for an SMBus segment and are not part of the translation circuit. They are shown for completeness.
- The gates of the FETs are tied to a voltage approximately  $V_{gs}$  above the lower rail voltage. A resistive divider is shown, but a convenient power rail would do nicely.
- Care must be taken to install the FETs so that any body diode does not conduct.
- The key requirement is that the high side drive low enough to register as a low on the low side. (High side  $V_{ol} < V_{il}$  on low side).

### 5.4 SB-TSI Registers

Reads to unimplemented registers return 00h. Writes to unimplemented registers are discarded.

**SBTSIx01 [CPU Integer Temperature] (SBTSI::CpuTempInt)**

Read-only.

The CPU temperature is calculated by adding the CPU temperature offset (SBTSI::CpuTempOffInt, SBTSI::CpuTempOffDec) to the processor control temperature (Tctl). SBTSI::CpuTempInt and SBTSI::CpuTempDec combine to return the CPU temperature. For the temperature encoding, see 5.2.5 [SB-TSI Temperature and Threshold Encodings]

Bits	Description
7:0	<b>CpuTempInt: integer CPU temperature value.</b> Read-only. Reset: Cold,XXh. This field returns the integer portion of the CPU temperature.

**SBTSIx02 [SB-TSI Status] (SBTSI::Status)**

Read-only, Volatile.

If SBTSI::AlertConfig[AlertCompEn] == 0, the temperature alert is latched high until the alert is Read. If SBTSI::AlertConfig[AlertCompEn] == 1, the alert is cleared when the temperature does not meet the threshold conditions for temperature and number of samples. See 5.2.3 [Alert Behavior].

Bits	Description
7:5	Reserved.
4	<b>TempHighAlert: temperature high alert.</b> Read-only, Volatile. Reset: Cold,X. 1=Indicates that the CPU temperature is greater than or equal to the high temperature threshold (SBTSI::HiTempInt, SBTSI::HiTempDec) for SBTSI::AlertThreshold[AlertThr] consecutive samples. 0=Indicates that the CPU temperature is less than the high temperature threshold (SBTSI::HiTempInt, SBTSI::HiTempDec) for SBTSI::AlertThreshold[AlertThr] samples and SBTSI::AlertConfig[AlertCompEn] == 1. Hardware will clear this bit when Read if SBTSI::AlertConfig[AlertCompEn] == 0.
3	<b>TempLowAlert: temperature low alert.</b> Read-only, Volatile. Reset: Cold,X. 1=Indicates that the CPU temperature is less than or equal to the low temperature threshold (SBTSI::LoTempInt, SBTSI::LoTempDec) for SBTSI::AlertThreshold[AlertThr] consecutive samples. 0=Indicates the CPU temperature is greater than the low temperature threshold (SBTSI::LoTempInt, SBTSI::LoTempDec) for SBTSI::AlertThreshold[AlertThr] samples and SBTSI::AlertConfig[AlertCompEn] == 1. Hardware will clear this bit when Read if SBTSI::AlertConfig[AlertCompEn] == 0.
2:0	Reserved.

**SBTSIx03 [SB-TSI Configuration] (SBTSI::Config)**

Reset: Cold,00h.

The bits in this register are Read-only and can be written by Writing to the corresponding bits in SBTSI::ConfigWr. See 5.2.3 [Alert Behavior] and 5.2.4 [Atomic Read Mechanism].

Bits	Description
7	<b>AlertMask: alert mask.</b> Read-only, Volatile. Reset: Cold,0. 0=ALERT_L pin enabled. 1=ALERT_L pin disabled and does not assert. IF (SBTSI::Config[AraDis] == 0) THEN Read-only; set-by-hardware. ELSE Read-only ENDIF. Hardware sets this bit if SBTSI::Config[AraDis] == 0, either SBTSI::Status[TempHighAlert] == 1 or SBTSI::Status[TempLowAlert] == 1, and a successful ARA is sent.
6	<b>RunStop: run stop.</b> Read-only. Reset: Cold,0. 0=Updates to SBTSI::CpuTempInt and SBTSI::CpuTempDec and the alert comparisons are enabled; Alert history counters (specified by SBTSI::AlertThreshold[AlertThr]) and the corresponding timer (specified by SBTSI::UpdateRate[UpRate]) continue to update. 1=Updates to SBTSI::CpuTempInt and SBTSI::CpuTempDec and the alert comparisons are disabled; Alert history counters (specified by SBTSI::AlertThreshold[AlertThr]) and the corresponding timer (specified by SBTSI::UpdateRate[UpRate]) are stopped. See 5.2.4 [Atomic Read Mechanism] for further details.
5	<b>ReadOrder: atomic read order.</b> Read-only. Reset: Cold,0. 0=Reading SBTSI::CpuTempInt causes the state of SBTSI::CpuTempDec to be latched. 1=Reading SBTSI::CpuTempDec causes the state of SBTSI::CpuTempInt to be latched. See 5.2.4 [Atomic Read Mechanism] for further details.
4:2	Reserved.
1	<b>AraDis: ARA disable.</b> Read-only. Reset: Cold,0. Read-only. 1=ARA response disabled.
0	Reserved.

**SBTSIx04 [Update Rate] (SBTSI::UpdateRate)**

Read-write. Reset: Cold,08h.

Bits	Description
7:0	<b>UpRate: update rate.</b> Read-write. Reset: Cold,08h. This field specifies the rate at which CPU temperature is compared against the temperature thresholds to determine if an alert event has occurred. Write access causes a reset of the alert history counters (specified by SBTSI::AlertThreshold[AlertThr]) and the corresponding timer (specified by SBTSI::UpdateRate[UpRate]).
<b>ValidValues:</b>	
Value	Description
00h	0.0625 Hz
01h	0.125 Hz
02h	0.25 Hz
03h	0.5 Hz
04h	1 Hz
05h	2 Hz
06h	4 Hz
07h	8 Hz
08h	16 Hz
09h	32 Hz
0Ah	64 Hz
FFh-0Bh	Reserved.

**SBTSIx07 [High Temperature Integer Threshold] (SBTSI::HiTempInt)**

Read-write. Reset: Cold,46h.

The high temperature threshold specifies the CPU temperature that causes ALERT\_L to assert if the CPU temperature is greater than or equal to the threshold. SBTSI::HiTempInt and SBTSI::HiTempDec combine to specify the high temperature threshold. See 5.2.5 [SB-TSI Temperature and Threshold Encodings]. Reset value equals 70 °C. Write access causes a reset of the alert history counters (specified by SBTSI::AlertThreshold[AlertThr]) and the corresponding timer (specified by SBTSI::UpdateRate[UpRate]). See 5.2.3 [Alert Behavior].

Bits	Description
7:0	<b>HiTempInt: high temperature integer threshold.</b> Read-write. Reset: Cold,46h. This field specifies the integer portion of the high temperature threshold.

**SBTSIx08 [Low Temperature Integer Threshold] (SBTSI::LoTempInt)**

Read-write. Reset: Cold,00h.

The low temperature threshold specifies the CPU temperature that causes ALERT\_L to assert if the CPU temperature is less than or equal to the threshold. SBTSI::LoTempInt and SBTSI::LoTempDec combine to specify the low temperature threshold. See 5.2.5 [SB-TSI Temperature and Threshold Encodings]. Write access causes a reset of the alert history counters (specified by SBTSI::AlertThreshold[AlertThr]) and the corresponding timer (specified by SBTSI::UpdateRate[UpRate]). See 5.2.3 [Alert Behavior].

Bits	Description
7:0	<b>LoTempInt: low temperature integer threshold.</b> Read-write. Reset: Cold,00h. This field specifies the integer portion of the low temperature threshold.

**SBTSIx09 [SB-TSI Configuration Write] (SBTSI::ConfigWr)**

Read-write. Reset: Cold,00h.

This register provides write access to SBTSI::Config.

Bits	Description
7	<b>AlertMask: alert mask.</b> Read-write. Reset: Cold,0. See SBTSI::Config[AlertMask].
6	<b>RunStop: run stop.</b> Read-write. Reset: Cold,0. See SBTSI::Config[RunStop].
5	<b>ReadOrder: atomic read order.</b> Read-write. Reset: Cold,0. See SBTSI::Config[ReadOrder].
4:2	Reserved.
1	<b>AraDis: ARA disable.</b> Read-write. Reset: Cold,0. See SBTSI::Config[AraDis].
0	Reserved.

**SBTSIx10 [CPU Decimal Temperature] (SBTSI::CpuTempDec)**

Read-only.

See SBTSI::CpuTempInt.

Bits	Description
7:5	<b>CpuTempDec: decimal CPU temperature value.</b> Read-only. Reset: Cold,XXXb. Read-only. This field returns the decimal portion of the CPU temperature.
4:0	Reserved.

**SBTSIx11 [CPU Temperature Offset High Byte] (SBTSI::CpuTempOffInt)**

Read-write. Reset: Cold,00h.

SBTSI::CpuTempOffInt and SBTSI::CpuTempOffDec combine to specify the CPU temperature offset. See 5.2.6 [SB-TSI Temperature Offset Encoding] for encoding details.

Bits	Description
7:0	<b>CpuTempOffInt: CPU temperature integer offset.</b> Read-write. Reset: Cold,00h. This field specifies the integer portion of the CPU temperature offset added to Tctl to calculate the CPU temperature. Write access causes a reset of the alert history counters (specified by SBTSI::AlertThreshold[AlertThr]) and the corresponding timer (specified by SBTSI::UpdateRate[UpRate]).

**SBTSIx12 [CPU Temperature Decimal Offset] (SBTSI::CpuTempOffDec)**

Read-write. Reset: Cold,00h.

See SBTSI::CpuTempOffInt.

Bits	Description
7:5	<b>CpuTempOffDec: CPU temperature decimal offset.</b> Read-write. Reset: Cold,0h. This field specifies the decimal/fractional portion of the CPU temperature offset added to Tctl to calculate the CPU temperature. Write access causes a reset of the alert history counters (specified by SBTSI::AlertThreshold[AlertThr]) and the corresponding timer (specified by SBTSI::UpdateRate[UpRate]).
4:0	Reserved.

**SBTSIx13 [High Temperature Decimal Threshold] (SBTSI::HiTempDec)**

Read-write. Reset: Cold,00h.

See SBTSI::HiTempInt.

Bits	Description
7:5	<b>HiTempDec: high temperature decimal threshold.</b> Read-write. Reset: Cold,0h. This field specifies the decimal portion of the high temperature threshold.
4:0	Reserved.

**SBTSIx14 [Low Temperature Decimal Threshold] (SBTSI::LoTempDec)**

Read-write. Reset: Cold,00h.

See SBTSI::LoTempInt.

Bits	Description
7:5	<b>LoTempDec: low temperature decimal threshold.</b> Read-write. Reset: Cold,0h. This field specifies the decimal portion of the low temperature threshold.
4:0	Reserved.

**SBTSIx22 [Timeout Configuration] (SBTSI::TimeoutConfig)**

Read-write. Reset: Cold,80h.

Bits	Description
7	<b>TimeoutEn: SMBus timeout enable.</b> Read-write. Reset: Cold,1. 0=SMBus defined timeout support disabled. 1=SMBus defined timeout support enabled. SMBus timeout enable.
6:0	Reserved.

**SBTSIx32 [Alert Threshold Register] (SBTSI::AlertThreshold)**

Read-write. Reset: Cold,00h.

See 5.2.3 [Alert Behavior].

Bits	Description
7:3	Reserved.
2:0	<b>AlertThr: alert threshold.</b> Read-write. Reset: Cold,0h. Specifies the number of consecutive CPU temperature samples for which a temperature alert condition needs to remain valid before the corresponding alert bit is set. For SBTSI::AlertConfig[AlertCompEn] == 1, it specifies the number of consecutive CPU temperature samples for which a temperature alert condition need to remain not valid before the corresponding alert bit gets cleared. Write access resets the alert history counters (specified by SBTSI::AlertThreshold[AlertThr]) and the corresponding timer (specified by SBTSI::UpdateRate[UpRate]). Details in SBTSI::Status.
<b>ValidValues:</b>	
<b>Value</b>	<b>Description</b>
0h	1 Sample
6h-1h	<Value+1> Samples
7h	8 Samples

**SBTSIxBF [Alert Configuration] (SBTSI::AlertConfig)**

Read-write.

Bits	Description
7:1	Reserved.
0	<b>AlertCompEn: alert comparator mode enable.</b> Read-write. Reset: Cold,X. 0=SBTSI::Status[TempHighAlert] and SBTSI::Status[TempLowAlert] are Read to clear. 1=SBTSI::Status[TempHighAlert] and SBTSI::Status[TempLowAlert] are Read-only; ARA response disabled. Write access does not change the alert history counters (specified by SBTSI::AlertThreshold[AlertThr]) or the corresponding timer (specified by SBTSI::UpdateRate[UpRate]). See SBTSI::Status.

**SBTSIxFE [Manufacture ID] (SBTSI::ManId)**

Read-only. Reset: Cold,00h.

Bits	Description
7:1	Reserved.
0	<b>ManId: Manufacture ID.</b> Read-only. Reset: Cold,0. Returns the AMD manufacture ID.

**SBTSIxFF [Revision] (SBTSI::Revision)**

Read-only. Reset: Cold,04h.

Bits	Description
7:0	<b>Revision: SB-TSI revision.</b> Read-only. Reset: Cold,04h. Specifies the SBI temperature sensor interface revision.



## 6 DXIO

### 6.1 DXIO Subsystem Overview

The DXIO subsystem provides a systematic infrastructure for sharing SERDES PHY I/Os among multiple serial protocols through a combined hardware and firmware solution. This block of logic consists of:

- PCS (Physical Coding Sub-layer) logic
- SERDES PHY (Serial/Deserial Physical Layer) logic
- A processor-wide, distributed, compressed-packet, crossbar/router responsible for mapping various protocol controllers to the PHY
- Topology management firmware

The DXIO subsystem provides a flexible platform-level solution for configuring AMD's processors to align to a variety of platform interface requirements.

#### 6.1.1 Definitions

*Table 88: Link Definitions*

Term	Description
<b>DXIO</b>	Distributed IO Crossbar Subsystem
<b>PMA</b>	Physical Media Access. a.k.a. PHY
<b>PIK</b>	PHY Interface Kompressor
<b>DPIK</b>	PHY Interface De-Kompressor
<b>KPRI</b>	Kompressed Physical-to-Raw PCS Interface
<b>KPMX</b>	KPRI multiplexing element
<b>KPFIFO</b>	KPRI Transmit Elasticity FIFO element
<b>KPNP</b>	Kompressed Packet Near PHY element
<b>GOP</b>	GMI over PCIe® link, referred to as the data link layer of either xGMI or WAFL
<b>GOPX1</b>	GMI over PCIe® link, with a link-width of 1 lane, referred to as WAFL link layer
<b>GOPX16</b>	GMI over PCIe® link, with a link-width of 16 lanes, referred to as xGMI data link layer

## 7 Fusion Controller Hub (FCH)

### 7.1 FCH Overview

#### 7.1.1 Acronyms

*Table 89: List of Acronyms used in FCH*

Acronym	Definition
ASF	Alert Standard Format.
ASL	ACPI Source Language. See docACPI.
DASH	Desktop and mobile Architecture for System Hardware.
SCH	Server Controller Hub.
TPM	Trusted Platform Module.

#### 7.1.2 Functional

This section describes the integrated FCH. It utilizes a standard Scalable Data Fabric Port (SDP).

The following is the list of IP blocks and functions:

LPC/SPI/eSPI – Low Pin Count/Serial peripheral interface: this is the bridge logic to the BIOS/firmware flash and SPI TPM. eSPI is multiplexed on the SPI bus to support an eSPI device such as embedded controller (EC). processors have the following features:

1. LPC function
2. The LPC is multiplexed on SPI pins

eMMC – Embedded MMC: a solid state flash interface that is popular for low cost embedded applications. The latest revision v5.0 supports bandwidths up to 400MBs.

CLKGEN – Clock Generation: integrated clock generation function for the entire system. This block adds a non-spread display and more PCIe® GPP clock outputs. It also adds more internal reference clocks for various PHYs. See 7.3.9.1 [Miscellaneous (MISC) Registers] for register descriptions.

GPIO – General Purpose IO: defined by AOAC and used as GPIO or interrupt inputs. See 7.3.10 [GPIO Pin control registers] for register descriptions.

ACPI – Advanced Configuration and Power Interface: power management and reset functions.

Power Management (PM) Supervisory – custom logic to manage clock and power gating to support AOAC. See 7.3.9.2 [Power Management (PM) Registers and Standard ACPI Registers] for register descriptions.

### 7.1.3 Memory Mapped IO (MMIO) Programming for Legacy Devices

The legacy devices, LPC, IOAPIC, ACPI, TPM, and Watchdog Timer, require the base address of the MMIO registers to be assigned before these registers can be accessed. The MMIO register base address and its entire range should be mapped to a non-posted memory region by programming the CPU register.

#### 7.1.3.1 Description for FCH::IO::PCIInterrupt Map

FCH::IO::pci\_intr\_index bit[7] means PciIntrApic, set as 0 means IRQ routing to PIC, set as 1 means IRQ routing to IOAPIC.

FCH::IO::pci\_intr\_index bits[6:0] are PCI interrupt index. Select which PCI interrupt to map. Following are detail description.

Table 90: ValidValuesTable: PCI interrupt index list of PCI\_INTR\_INDEX bit[6:0]

Value	Description
04h-00h	INT[E:A]#.
05h	INTF#/GENINT2.
06h	INTG#.
07h	INTH#.
08h	Misc.
0Bh-09h	Misc[2:0].
0Fh-0Ch	INT[D:A] from serial IRQ.
10h	SCI.
11h	SMBUS0.
12h	ASF.
15h-13h	Reserved.
16h	PerMon.
19h-17h	Reserved.
1Ah	SDIO.
1Fh-1Bh	Reserved.
20h	CIR, no IRQ connected.
21h	GPIOa, from PAD_FANIN0.
22h	GPIOb, from PAD_FANOUT0.
23h	GPIOc, no IRQ connected.
5Fh-24h	Reserved.
60h	GEventSci Interrupt.
61h	GEventSmi Interrupt.
62h	GPIO Controller Interrupt.
6Fh-63h	Reserved.
70h	I2C0.
71h	I2C1.
72h	I2C2.
73h	I2C3.
75h-74h	UART1/UART0.
76h	Reserved.
77h	UART4.
79h-78h	UART3/UART2.
FFh-7Ah	Reserved.

### 7.1.4 EMMC

The eMMC (embedded MultiMedia Card controller) is controlled by driver software. There are no programming requirements for the BIOS. The eMMC interface pins can be either 1.8V or 3.3V.

### 7.1.5 eSPI

eSPI is configured to run at 16.6MHz, 20MHz (default), 25MHz, 33MHz, 50MHz and 66MHz. See 7.3.8.1 [eSPI Registers] about how to program the speed.

### 7.1.6 I3C Master

The I3C interface is improved from I2C interface and preserves the backward compatibility with I2C. Following are the supporting features:

1. Two wire serial interface up to 12.5MHz using Push-Pull
2. Legacy I2C Device co-existence on the same Bus (with some limitations)
3. Dynamic Addressing while supporting Static Addressing for Legacy I2C Devices
4. Broadcast and Direct Common Command Code (CCC) message
5. Legacy I2C messaging
6. Support Single Data Rate messaging (SDR)
7. In-Band Interrupt support
8. Hot-Join support
9. Master mode only

For I3C0~I3C3, the I3C pins interface pins can be either at 1.1V or 1.8V.

*Table 91: I3C PAD Control Configuration*

I3C modules	Description	Register Location
I3C_0	Pad control Register	FCH::MISC::i2c0_padctrl
I3C_1	Pad control Register	FCH::MISC::i2c1_padctrl
I3C_2	Pad control Register	FCH::MISC::i2c2_padctrl
I3C_3	Pad control Register	FCH::MISC::i2c3_padctrl

### 7.1.7 GPIO

There are four groups of GPIO, GPIO Bank0 ~ Bank3. See 7.3.10.2 [GPIO Registers].

To enable the GPIO controller, follow the programming steps listed below:

#### 7.1.7.1 Interrupt GPIO

1. The driver programs the DebounceTmrOut/DebounceTmrOutUnit/DebounceCntrl bits according to the data passed by BIOS through the ASL code.
2. Configure bit[10:8] according to the data passed by BIOS through ASL code.
3. Set bit[11] and bit[12] to be 1 to enable interrupt delivery and interrupt status. Interrupt status is at bit[28] when the GPIO input is asserted.
4. Configure the GPIO controller interrupt. The GPIO controller interrupt is an active low level interrupt. The driver clears the interrupt status, then sets FCH::GPIO::gpio\_wake\_interrupt\_master\_switch[eoi] = 1 to de-assert the interrupt request after acknowledging the interrupt.

### 7.1.7.2 Wakeup GPIO

1. BIOS programs bit[15:13] to enable wake in S0i3/S3/S5 by BIOS. The wake status is bit[29]. BIOS/SMU reads the wake status to determine the wake source, then clears them after wake.
2. The driver programs DebounceTmrOut/DebounceTmrOutUnit/DebounceCntl bits according the data from BIOS through the ASL code.
3. The driver sets bit[10:8] to configure assertion condition according to the data from BIOS through the ASL code.
4. Set bit[11] and bit[12] to enable interrupt delivery and interrupt status then its interrupt status is found at bit[28] when the GPIO input is asserted.
5. Configure the GPIO controller interrupt. The GPIO controller interrupt is an active low level interrupt. BIOS configures it and sets it to 0s. The driver clears the interrupt status then sets FCH::GPIO::gpio\_wake\_interrupt\_master\_switch[eo] = 1 after the interrupt is acknowledged.

### 7.1.7.3 Pure GPIO

Bit[23:17] provides the state of the control on pull-up, pull-down, drive strength, output enable, output value, and pin value through bit[16].

### 7.1.8 Sticky/non-sticky attribute of PMIO registers

Note: when S5 PMIO non-sticky fields are programmed, notice that S0 and S5 registers' read-back values are different on S0i3/Z9/Z10 exit. The value when you read back in S0 is the register's reset value, while the value you read back in S5 is the programmed value. There is quite short latency during S0i3/Z9/Z10 exit, so

FCH::AOAC::s0i3\_debug[s0i3\_dontblockpmiorst] is designed to block S5 PciRstB in S0i3/Z9/Z10 exit by default.

The table below lists the sticky/non-sticky attribute of all PMIO registers. Sticky/non-sticky attribute is directly related to the registers' reset condition.

1. A "Sticky" register is reset in the following conditions:
  1. Resume Reset: This reset is asserted in G3 state, and deasserted during G3 to S5 transition.
  2. System Reset: From system reset button.
2. A "non-sticky" register is reset in the following conditions:
  1. Resume Reset: This reset is asserted in G3 state, and deasserted during G3 to S5 transition.
  2. System Reset: From system reset button.
  3. S0 Reset events: Some events that happen in S0 state, such as CF9 and Keyboard Reset.
  4. Traditional sleep: S3 and S5 states.

PMIO registers information is in 7.3.9.2 [Power Management (PM) Registers and Standard ACPI Registers].

*Table 92: PMIO Register Sticky/Non-sticky Attributes*

Dword#	Offset	Sticky/Non-sticky
0	00h ~ 03h	Sticky
1	04h ~ 07h	[31:17] Sticky [16:15] non-sticky [14: 0 ] Sticky
2	08h ~ 0Bh	Sticky
3	0Ch ~ 0Fh	Sticky
4	10h ~ 13h	Sticky
5	14h ~ 17h	Sticky
6	18h ~ 1Bh	Sticky
7	1Ch ~ 1Fh	Sticky
8	20h ~ 23h	Sticky [1] only reset by s5_rst

9	24h ~ 27h	Sticky
10	28h ~ 2Bh	Sticky
11	2Ch ~ 2Fh	Sticky
12	30h ~ 33h	Sticky
13	34h ~ 37h	Sticky
14	38h ~ 3Bh	Sticky
15	3Ch ~ 3Fh	Sticky
16	40h ~ 43h	Sticky
17	44h ~ 47h	[31] non-sticky [30] sticky, reset by s5 [29] non-sticky [28:25] sticky [24:0] read-only
18	48h ~ 4Bh	Sticky
19	4Ch ~ 4Fh	Sticky
20	50h ~ 53h	Sticky
21	54h ~ 57h	[31:8] Sticky [7] Sticky when [14]=0, non-sticky when [14]=1 [6:0] Sticky
22	58h ~ 5Bh	[31:28] no reset (these bits are actually in RTC) [27:23] sticky [22] read-only [21] write-only, read always return 0, tie 0 when read [20:19] sticky [18] read-only [17] sticky [16:0] sticky
23	5Ch ~ 5Fh	[31:24] no reset (these bits are actually in RTC) [23:11] sticky [10] reserved, read always return 1, tie 1 at s5 power [9:0] sticky
24	60h ~ 63h	Sticky
25	64h ~ 67h	Sticky
26	68h ~ 6Bh	Sticky
27	6Ch ~ 6Fh	Sticky
28	70h ~ 73h	Sticky
29	74h ~ 77h	[31] sticky, need to be reset by rsmrst [30:17] sticky [16] sticky, reset to 0 by PLLock=0 [15:8] sticky [7] write-only, read always return 0 [6:0] sticky
30	78h ~ 7Bh	Non-sticky
31	7Ch ~ 7Fh	Non-sticky
32	80h ~ 83h	Sticky
33	84h ~ 87h	Non-sticky
34	88h ~ 8Bh	Non-sticky
35	8Ch ~ 8Fh	[31:29] sticky [28] sticky, need be reset by rsmrst [27:0] Sticky

36	90h ~ 93h	Sticky
37	94h ~ 97h	Sticky
38	98h ~ 9Bh	Sticky
39	9Ch ~ 9Fh	Read-only, only this one is true read only
40	A0h ~ A3h	non-sticky
41	A4h ~ A7h	[31] non-sticky [30:29] reserved, read always return 0 [28:0] is read-only if [31] is 1, otherwise it is non-sticky
42	A8h ~ Abh	Sticky
43	Ach ~ Afh	Reserved, read always return 0, can't write, read tie 0
44	B0h ~ B3h	[31:28] non-sticky [27:23] read-only [22:0] non-sticky
45	B4h ~ B7h	Sticky
46	B8h ~ BBh	[31:21] sticky [20] read-only [19:0] sticky
47	BCh ~ BFh	[31:8]Sticky [7]non-sticky [6:0]Sticky
48	C0h ~ C3h	[31:30] sticky [29:0] is sticky if 0xC4[2] is 1, otherwise it is read-only
49	C4h ~ C7h	[31:12] sticky [11:9] can write, read return Shawdowcf9 [8] reserved, read always return 0 [7:1] sticky [0] reserved, read always return 0
50	C8h ~ CBh	Sticky
51	CCh ~ CFh	Sticky
52	D0h ~ D3h	[31:16] sticky [15:8] sticky, but is indirect access to four 8-bit registers [7:0] sticky
53	D4h ~ D7h	[31:8]Sticky, can be locked by bit10. Once locked, can't write [7:0]Sticky
54	D8h ~ DBh	sticky
55	DCh ~ DFh	Sticky
56	E0h ~ E3h	Sticky
57	E4h ~ E7h	Sticky
58	E8h ~ Ebh	Sticky
59	Ech ~ Efh	[31:20] sticky [19] read-only [18:0] sticky [5]reset by rsmrst
60	F0h ~ F3h	Sticky
61	F4h ~ F7h	Sticky
62	F8h ~ FBh	Sticky
63	FCh ~ FFh	Sticky

## 7.1.9 CLK IP Overview

### 7.1.9.1 iCLK/eCLK mode

There are two clocking modes in which the processor can be brought up based on the boot strap pin SPI\_CLK0: iCLK and eCLK clocking modes. iCLK clocking mode means that chip takes an internal crystal generated clock as reference clock while eCLK clocking mode means that chip takes an external clock as reference clock.

BIOS should read FCH::MISC::strapstatus[clkgenstrap] to determine the clocking mode.

### 7.1.9.2 CGPLL SSC Enable/Disable

The CGPLL SSC is disabled by default.

In eCLK mode, Spread Spectrum Clocking (SSC) should remain disabled.

In iCLK mode, SSC can be enabled on the CGPLL. The CGPLL default spread amount is -0.375%. Alternatively, the spread amount can be set to -0.3%.

SSC enable sequence only need be executed during initial boot-up sequence or reset sequence triggered by pushing reset button, that means, for example, this sequence should not be executed after system exit Sleep State.

Note: When spread spectrum is turned on, the effective clock frequency will be slowed down by the same amount. This is applicable to functions that derive clocks from this PLL. For example, the effective CPU core frequency will be slowed down approximately  $\frac{1}{2}$  of the spread amount.

#### 7.1.9.2.1 CGPLL -0.375% SSC Programming

Software can enable the CGPLL default -0.375% SSC setting by the following programming:

1. FCH::MISC::cgpllconfig1[cg1\_spread\_spectrum\_enable] = 1
2. FCH::MISC::cgpllconfig3[cg1pll\_fracn\_en\_override] = 1
3. FCH::MISC::miscclkcntrl0[cg1\_cfg\_update\_req] = 1

Software can disable the CGPLL default -0.375% SSC setting by the following programming:

1. FCH::MISC::cgpllconfig1[cg1\_spread\_spectrum\_enable] = 0
2. FCH::MISC::miscclkcntrl0[cg1\_cfg\_update\_req] = 1

#### 7.1.9.2.2 CGPLL -0.3% SSC Programming

Software can enable the CGPLL -0.3% SSC setting by the following programming:

1. FCH::MISC::cgpllconfig4[cg1pll\_fcw1\_frac\_override] = 16'h1333
2. FCH::MISC::cgpllconfig5[cg1pll\_fcw\_slew\_frac\_override] = 16'h003A
3. FCH::MISC::cgpllconfig1[cg1\_spread\_spectrum\_enable] = 1'b1
4. FCH::MISC::cgpllconfig3[cg1pll\_fracn\_en\_override] = 1'b1
5. FCH::MISC::miscclkcntrl0[cg1\_fbdiv\_loaden] = 1'b1
6. FCH::MISC::miscclkcntrl0[cg1\_cfg\_update\_req] = 1'b1

Software can disable the CGPLL -0.3% SSC setting by the following programming:

1. FCH::MISC::cgpllconfig4[cg1pll\_fcw1\_frac\_override] = 16'h0000
2. FCH::MISC::cgpllconfig5[cg1pll\_fcw\_slew\_frac\_override] = 16'h0000



3. FCH::MISC::cgpllconfig1[cg1\_spread\_spectrum\_enable] = 1'b0
4. FCH::MISC::miscclkcntrl0[cg1\_fbdiv\_loaden] = 1'b0
5. FCH::MISC::miscclkcntrl0[cg1\_cfg\_update\_req] = 1'b1

### 7.1.9.3 Global A-Link/B-Link Clock Gating

A-Link and B-Link clocks are 2 global clocks used inside FCH for most of sub-IPs.

Software enables the global A-Link clock gate off function by programming:

1. FCH::MISC::clkgatedcntl[alinkclk\_gateoffen] = 1
2. FCH::PM::isacontrol[abclkgateen] = 1

Software enables the global B-Link clock gate off function by programming:

1. FCH::MISC::clkgatedcntl[blinkclk\_gateoffen] = 1
2. FCH::PM::isacontrol[abclkgateen] = 1

FCH::MISC::clkgatedcntl[alinkclk\_gateoffen] and FCH::MISC::clkgatedcntl[blinkclk\_gateoffen] are sticky bits. FCH::PM::isacontrol[abclkgateen] is a non-sticky bit that should be re-programmed to 1 after resets defined in 7.1.8 [Sticky/non-sticky attribute of PMIO registers] if global A-Link/B-Link gating functions been enabled.

Note: to enable FCH\_SDP disconnect to allow LCLK deep sleep.

### 7.1.9.4 GPP CLKREQB Mapping Table

Table 93: GPP ClkREQB Mapping

ClkReq Map Register	Package Pin Name
FCH::MISC::gppclkcntrl[gpp_clk0_clock_request_mapping]	BP_CLK_REQ0_L
FCH::MISC::gppclkcntrl[gpp_clk1_clock_request_mapping]	BP_CLK_REQ1_L/AGPIO115
FCH::MISC::gppclkcntrl[gpp_clk2_clock_request_mapping]	BP_CLK_REQ2_L/AGPIO116
FCH::MISC::gppclkcntrl[gpp_clk3_clock_request_mapping]	BP_CLK_REQ3_L
FCH::MISC::gppclkcntrl[gpp_clk4_clock_request_mapping]	BP_CLK_REQ4_L/OSCIN/EGPIO132
FCH::MISC::gppclkcntrl[gpp_clk5_clock_request_mapping]	BP_CLK_REQ5_L/AGPIO38
FCH::MISC::gppclkcntrl[gpp_clk6_clock_request_mapping]	BP_CLK_REQ6_L/AGPIO39

### 7.1.10 FCH Register Access Information Guide

Following registers can only be accessed in 8-bit (byte access), using 16-bit (word access) or 32-bit (double word access) will have quirky behavior. For Read, byte0 data will be returned on all 4 bytes, for Write, only byte0 data will be written.

- IOMUX: Memory mapped address 0xFED8\_0D00 – 0xFED8\_0DFF.
- BIOS\_RAM: Access using IO (0xCD4: Index, 0xCD5: Data) or Memory mapped address 0xFED8\_0500 – 0xFED8\_05FF.
- CMOS\_RAM: Access using IO (0x72: Index, 0x73: Data) or Memory mapped address 0xFED8\_0600 – 0xFED8\_06FF.
- CMOS: Access using IO (0x70: Index, 0x71: Data) or Memory mapped address 0xFED8\_0700 – 0xFED8\_07FF.
- PMIO2: Memory mapped address 0xFED8\_0400 – 0xFED8\_04FF.
- ACPI: registers can be 8/16/32-bit, please refer to following table. AcpiMMioAddr=0xFED8\_0000.

Table 94: Register Access Information

Register Name	IO Base Address definition register	IO Offset Address*	MMIO Access
Pm1Status	PM_60: AcpiPm1EvtBlk	00h, 16-bit	AcpiMMioAddr + 800
Pm1Enable		02h, 16-bit	AcpiMMioAddr + 802
PmControl	PM_62: AcpiPm1CntBlk	00h, 16-bit	AcpiMMioAddr + 804
TmrValue/ETmrValue	PM_64: AcpiPmTmrBlk	00h, 32-bit, Read Only	AcpiMMioAddr + 808
CLKVALUE	PM_66: CpuControl	00h, 32-bit	AcpiMMioAddr + 80C
PLvl2		04h, 8-bit, Read Only	AcpiMMioAddr + 810
PLvl3		05h, 8-bit, Read Only	AcpiMMioAddr + 811
EVENT_STATUS_STDACPI	PM_68: AcpiGpe0Blk	00h, 32-bit	AcpiMMioAddr + 814
EVENT_ENABLE_STDACPI		04h, 32-bit	AcpiMMioAddr + 818
SmiCmdPort	PM_6A: AcpiSmiCmd	00h, 8-bit	AcpiMMioAddr + 81C
SmiCmdStatus		01h, 8-bit	AcpiMMioAddr + 81D
PmaControl	PM_6E: AcpiPmaCntBlk	00h, 8-bit	AcpiMMioAddr + 824
PmaControl_1		04h, 8-bit	AcpiMMioAddr + 828

Note: The offset addresses listed here for the ACPI registers belong to different apertures/decodes. Check the register descriptions for details.

### 7.1.11 Reset Overview

Below are definitions of the various reset types:

- Type 0 reset (S5 Reset): RsmRst and UserRst.
- Type 1 reset (reset initiated by software or system): CF9, KBRst, Sync\_flood, ASF\_remote\_reset, Fail\_boot, Watchdog Timer reset, toggling of PwrGood (SLP\_S3#/SLP\_S5# remain de-asserted at high), SHUTDOWN command, INIT/PORT92.
- Type 2 reset (Sleep Reset): S3/S4/S5 reset.
- Type 3 reset (Fatal\_error\_reset or reset caused by hardware exception): 4s-shutdown, thermal trip, ASF\_remotePowerDown.
- Type 4 reset (any reset from above): Type 0 or Type 1 or Type 2 or Type 3.

Table 95: Reset Type

Register block	Power domain	Reset source
PCI Configure	S0	Type 4
SMI (7.3.3 [SMI Registers])	S5	Some register: Type 0 or Type 3 Some register: Type 4
PM (7.3.9.2 [Power Management (PM) Registers]	S5	Some register: Type 0

and Standard ACPI Registers])		Some register: Type 4
PMIO2 (7.3.9.4 [Power Management (PM2) Registers])	S5	FCH::PM::resetcontrol 1[rsttocpupwrgden] == 1 ? Type 4 : Type 0
ACPI (7.3.9.2 [Power Management (PM) Registers and Standard ACPI Registers])	S5	Type 0 or Type 3
ASF	S5	Type 0
SMBUS	S5	
WatchDog (7.3.5 [Watchdog Timer (WDT) Registers])	S5	Type 0
HPET (7.3.4 [High Precision Event Timer (HPET) Registers])	S0	Type 4
IOMUX (7.3.10.1 [IOMUX Registers])	S5 (0-42) S0 (67-148)	S5-IoMux: Type 0 S0-IoMux: Type 4
MISC (7.3.9.1 [Miscellaneous (MISC) Registers])	S5	Most registers are Type 0. Some CLK register are Type 4 (i.e., FCH::MISC::miscclkcntl3, FCH::MISC::autoaddrflow, ...)
Serial Debug	S5	Type 0
Shadow System Counter	S5	Type 0 or Type 1
GPIO-0 (7.3.10.2 [GPIO Registers])	S5	Type 0 or Type 1
GPIO-1 (7.3.10.2 [GPIO Registers])	S0	Type 4
GPIO-2 (7.3.10.2 [GPIO Registers])	S0	Type 4
GPIO-3 (7.3.10.2 [GPIO Registers])	S0	Type 4
GPIO-4 (7.3.10.2 [GPIO Registers])	S0	Type 4
Wake Alarm (ACDC timer) (7.3.6 [Wake Alarm Device (AcDcTimer) Registers])	S5	Type 0
AOAC (7.3.7 [Always On Always Connected (AOAC) Registers])	S5	Type 0 or Type 1

Below is a set of simplified top level reset paths.

When a reset is generated from acpi\_s5, all S0 logic within the FCH are reset by it. It also resets devices on the motherboard that use PCIE\_RST\_L or LPC\_RST\_L. Other IPs within the SOC are not reset by it directly. Instead, they are being reset by SMU. when CpuPwrGood is connected to the SMU and the SMU treats it as an interrupt. Upon assertion of this signal, the SMU proceeds to do its "house cleaning" activities first; then it will issue resets to all other IPs that are not inside of the FCH. CpuPwrGood is the actual reset to the SMU. When this signal is de-asserted (to low), the SMU will propagate its reset to all IPs within the SOC. Warm reset is defined as an assertion of CpuRstB and cold reset the toggling of both CpuRstB (low) and CpuPwrGood (low).

Table 96: Reset Overview

S5 reset	Behavior
RSMRST_L	Reset all logic (S0 and S5). In addition, SOC will go back to default power state (always-off, always-on, previous state).
SYS_RST_L	Reset all logic (S0 and S5).
Sleep induced reset	Behavior
SLEEP3/4/5	Sleep entry to S3/4/5 will assert cold reset to all S0 logic prior to assertion of SLP_S3_L/SLP_S5_L.
Pin included reset	Behavior
PWRBUTTON_L	4 second override – unconditional shutdown (Sleep S5) → generates cold

	reset and will reset all S0 logic.
THERMTRIP_L	Over temperature indicator from CPU – unconditional shutdown (Sleep S5) → generates cold reset and will reset all S0 logic.
KB_RST_L	Keyboard reset → reset to S0 logic; configurable to warm or cold reset.
PWRGOOD	De-assertion of PWRGOOD will always generate cold reset to S0 logic.
Software induced reset	Behavior
CF9	Write to CF9 → reset S0 logic; configurable to warm or cold reset; or momentary S0 → S5 → S0 transition.
Port92	Write Port 92 (FAST_INIT) → reset S0 logic; configurable to warm or cold reset.
Software writes to PM_Reg 0xC4[6], resetallacpi	Generates reset to S0 and S5 logic.
Hardware based reset	Behavior
Watchdog timer	Watchdog timer → reset S0 logic; configurable to warm or cold reset.
AMD (boot_timer) Watchdog timer	AMD defined Watchdog timer → reset S0 logic; configurable to warm or cold reset.
Remote Command reset	Behavior
ASF – remote reset	Reset S0 logic; configurable to warm or cold reset.
ASF – remote sleep	Remote sleep S5 command – logic will sequence to S5 and assert cold reset to all S0 logic.
Internal events	Behavior
Shutdown (message from CPU)	Triple faults in CPU will cause an internal SHUTDOWN message to be broadcasted. FCH will generate a reset to S0 logic; configurable to warm or cold reset.
SYNC_FLOOD (message)	Internal data fabric logic detects an error (e.g., parity error) and broadcasts an internal SYNC_FLOOD message. FCH will generate a reset to S0 logic; configurable to warm or cold reset.

## 7.1.12 RTC

### 7.1.12.1 RTC Register

Please note that, the value in Time/Alarm or CMOS registers or CMOS RAM is undefined/indeterministic when power up first time. It is highly recommended to add a checksum or CRC over CMOS RAM so BIOS can detect whether CMOS has been corrupted or erased.

You can see RTC registers at 7.3.1 [Legacy Block Configuration Registers (IO)].

### 7.1.12.2 External RTC Decode

To support external RTC, using eSPI bus to decode RTC instead of using ACPI bus.

1. Program FCH::PM::rtcccontrol[rtcdecodedis] to 1'b1 to disable the integrated RTC decoding. The default value for this bit is disable (1'b0) which means integrated RTC decode is working.
2. Config ESPI IO space to decode for RTC ports. Related registers:
  1. LEGACYIOx00000070: FCH::IO::nmi\_enable
  2. LEGACYIOx00000071: FCH::IO::rtcdataport
  3. LEGACYIOx00000072: FCH::IO::alternatrtcaddrport
  4. LEGACYIOx00000073: FCH::IO::alternatrtcdataport

Note: the decode of FCH::IO::nmi\_enable[nmienableb] is also blocked by FCH::PM::rtcccontrol[rtcdecodedis]. Therefore, firmware has to config FCH::IO::nmi\_enable[nmienableb] before disabling integrated RTC decoding.

### 7.1.13 Strap Definition

Table 97: Strap Pins/Function

Ball Name	Strap Name	Type	Default Value	Description
BP_SPI_CLK0	CLKGEN	II	NA	Defines clock generator. "0" – Use 100Mhz PCIE clock as reference clock and generate internal clocks only. Note a 48Mhz crystal is still required in this configuration. "1" – Use 48Mhz crystal clock and generate both internal and external clocks. External pull-up/Pull-down is required on the pin.
BP_SYS_RESET_L	ShortReset	I	1	1: normal reset mode. 0: short reset mode. External pull-up/Pull-down is required on the pin.
BP_PKGID2	PKGID2	II	NA	Originally AGPIO86, used as PKGID bit2 and can't be used as GPIO anymore.
BP_PKGID0	PKGID0	II	NA	New pad, used as PKGID bit0.
BP_SPI1_CLK, BP_SPI2_CLK	ROMTYPE[1:0]	II	NA	00: SPI ROM 01: Reserved 10: SAFS 11: eSPI  The strap pin should be configured to the corresponding state that matches the hardware ROM type installed. External pull-up/Pull-down is required on the pin.

Even though some of the pins may have integrated pull-up enabled, external pull-up/down are required on all strap pins except for the two pins explicitly mentioned above

Type I strap: Use BP\_RSMRST\_L rising edge to latch the strap value. The strap pad can only locate in S5.

Type II strap: Use BP\_PWR\_GOOD as source and after S0 power up, count another counter (16 osc clock cycles), then latch the strap done. All straps on S0 pads use this type of strap.

### 7.1.14 Address Mapping Table

Table 98: Address Space Mapping under APB BUS

Function name	Address Mapping
I2C_0	0xFEDC_2xxx
I2C_1	0xFEDC_3xxx
I2C_2	0xFEDC_4xxx
I2C_3	0xFEDC_5xxx
DMA_0	0xFEDC_7xxx
DMA_1	0xFEDC_8xxx
UART_0	0xFEDC_9xxx
UART_1	0xFEDC_Axxx
DMA_2	0xFEDC_Cxxx
DMA_3	0xFEDC_Dxxx

UART_2	0xFEDC_Exxx
UART_3	0xFEDC_Fxxx
I3C_0	0xFEDD_2xxx
I3C_1	0xFEDD_3xxx
I3C_2	0xFEDD_4xxx
I3C_3	0xFEDD_6xxx
ACPI	0xFED8_0000 8KB memory Space
ACPI	0xFD_F95F_FF00 256B memory Space
SPI	0xFD_F95F_FE00 256B memory Space
SPI ROM	Top of 4G & 1M; 256B memory Space

## 7.2 FCH BTS Check Table

Table 99: FCH BTS Common Table

RegisterField	Type	Content
FCH::MISC::clkgatedcntl[blinkclk_gateoffen]	Check	1b
FCH::MISC::clkgatedcntl[alinkclk_gateoffen]	Check	1b

## 7.3 Registers

### 7.3.1 Legacy Block Configuration Registers (IO)

#### 7.3.1.1 Registers

IOx00C01_x00 [PCI INT[H#,G#,F#,E#,D#,C#,B#,A#] Map] (FCH::IO::PciIntMap)	
Read-write. Reset: 1Fh.	
_aliasIO; IOx00C01_x00; IO=0000_0000h; DataPortWrite=FCH::IO::pci_intr_index	
Bits	Description
7:5	Reserved.
4:0	<b>Pci2IntrMap</b> . Read-write. Reset: 1Fh. If (FCH::IO::pci_intr_index[7] == 1) then Pci2IntrMap specifies mapping of INT[H#:A#] to APIC interrupt number. If (FCH::IO::pci_intr_index[7] == 0) then Pci2IntrMap specifies mapping of INT[H#:A#] to PIC interrupt number.

**IOx00C01\_x08 [Intr\_Misc\_Map] (FCH::IO::IntrMiscMap)**

Read-write. Reset: 00h.

\_aliasIO; IOx00C01\_x08; IO=0000\_0000h; DataPortWrite=FCH::IO::pci\_intr\_index

Bits	Description										
7:6	<b>Pci2Intr15Map.</b> Read-write. Reset: 0h. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>IRQ15 mapped to legacy IDE.</td></tr> <tr> <td>1h</td><td>IRQ15 mapped to SATA IDE.</td></tr> <tr> <td>2h</td><td>IRQ15 mapped to SATA2.</td></tr> <tr> <td>3h</td><td>IRQ15 come from Serial IRQ or PCI interrupt.</td></tr> </table>	Value	Description	0h	IRQ15 mapped to legacy IDE.	1h	IRQ15 mapped to SATA IDE.	2h	IRQ15 mapped to SATA2.	3h	IRQ15 come from Serial IRQ or PCI interrupt.
Value	Description										
0h	IRQ15 mapped to legacy IDE.										
1h	IRQ15 mapped to SATA IDE.										
2h	IRQ15 mapped to SATA2.										
3h	IRQ15 come from Serial IRQ or PCI interrupt.										
5:4	<b>Pci2Intr14Map.</b> Read-write. Reset: 0h. <b>ValidValues:</b> <table> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>0h</td><td>IRQ14 mapped to legacy IDE.</td></tr> <tr> <td>1h</td><td>IRQ14 mapped to SATA IDE.</td></tr> <tr> <td>2h</td><td>IRQ14 mapped to SATA2.</td></tr> <tr> <td>3h</td><td>IRQ14 mapped to Serial IRQ or PCI interrupt.</td></tr> </table>	Value	Description	0h	IRQ14 mapped to legacy IDE.	1h	IRQ14 mapped to SATA IDE.	2h	IRQ14 mapped to SATA2.	3h	IRQ14 mapped to Serial IRQ or PCI interrupt.
Value	Description										
0h	IRQ14 mapped to legacy IDE.										
1h	IRQ14 mapped to SATA IDE.										
2h	IRQ14 mapped to SATA2.										
3h	IRQ14 mapped to Serial IRQ or PCI interrupt.										
3	<b>PciIntrIrq12.</b> Read-write. Reset: 0. 0=IMC as IRQ12 input source. 1=Serial IRQ or PCI devices as IRQ12 input source.										
2	<b>PciIntrIrq8.</b> Read-write. Reset: 0. 0=RTC is IRQ8 input source. 1=Serial IRQ or PCI devices as IRQ8 input source.										
1	<b>PciIntrIrq1.</b> Read-write. Reset: 0. 0=IMC as IRQ1 input source. 1=Serial IRQ or PCI devices as IRQ1 input source.										
0	<b>PciIntrIrq0.</b> Read-write. Reset: 0. 0=8254 timer as IRQ0 input source. 1=Serial IRQ or PCI devices as IRQ0 input source.										

**IOx00C01\_x09 [Intr\_Misc0Map] (FCH::IO::IntrMisc0Map)**

Read-write. Reset: E7h.

\_aliasIO; IOx00C01\_x09; IO=0000\_0000h; DataPortWrite=FCH::IO::pci\_intr\_index

Bits	Description
7	<b>IntrDelay.</b> Read-write. Reset: 1. INTR 600 ns delay.
6	<b>IRQ12FilterEnable.</b> Read-write. Reset: 1. IRQ12 filter enable.
5	<b>IRQ1FilterEnable.</b> Read-write. Reset: 1. IRQ1 filter enable.
4	<b>IrqInputEn.</b> Read-write. Reset: 0. 0=Mask off IRQ input. 1=Enable IRQ input.
3	<b>MaskIrq1Irq12.</b> Read-write. Reset: 0. 0=Enable IRQ1 and IRQ12. 1=Mask off IRQ1 and IRQ12.
2	<b>Merge_Ec_irq12.</b> Read-write. Reset: 1. 0=Route serial IRQ12 to USB IRQ12 input. 1=Route IMC IRQ12 to USB IRQ12 input.
1	<b>Merge_Ec_irq1.</b> Read-write. Reset: 1. 0=Route serial IRQ1 to USB IRQ1 input. 1=Route IMC IRQ1 to USB IRQ1 input.
0	<b>IntMap.</b> Read-write. Reset: 1. 0=INT0 in IOAPIC comes from IRQ0 in PIC, INT2 in IOAPIC comes from INTR in PIC. 1=INT2 in IOAPIC comes from IRQ0 in PIC, INT0 in IOAPIC comes from INTR in PIC.

**IOx00C01\_x0A [Intr\_Misc1Map] (FCH::IO::IntrMisc1Map)**

Read-write. Reset: 00h.

\_aliasIO; IOx00C01\_x0A; IO=0000\_0000h; DataPortWrite=FCH::IO::pci\_intr\_index

Bits	Description
7:0	<b>HPET.</b> Read-write. Reset: 00h. Writes to this register update the bits in FCH::TMR::HPET::tmr0_conf_cap_h [7:0], FCH::TMR::HPET::tmr1_conf_cap_h [7:0] and FCH::TMR::HPET::tmr2_conf_cap_h [7:0]; All 3 registers, HPETx104[7:0], HPETx124[7:0], and HPETx144[7:0], are updated at the same time. FCH::IO::IntrMisc1Map updates the lower 8 bits. FCH::IO::IntrMisc2Map updates the upper 8 bits.

**IOx00C01\_x0B [Intr\_Misc2Map] (FCH::IO::IntrMisc2Map)**

Read-write. Reset: 00h.

\_aliasIO; IOx00C01\_x0B; IO=0000\_0000h; DataPortWrite=FCH::IO::pci\_intr\_index

Bits	Description
7:0	<b>HPET</b> . Read-write. Reset: 00h. Writes to this register update this bits in FCH::TMR::HPET::tmr0_conf_cap_h [15:8], FCH::TMR::HPET::tmr1_conf_cap_h [15:8] and FCH::TMR::HPET::tmr2_conf_cap_h [15:8]; All 3 registers, HPETx104[15:8], HPETx124[15:8], and HPETx144[15:8], are updated at the same time. FCH::IO::IntrMisc1Map updates the lower 8 bits. FCH::IO::IntrMisc2Map updates the upper 8 bits.

**IOx00C01\_x0C [PCIInterruptMap] (FCH::IO::PCIInterruptMap)**

Read-write. Reset: 1Fh.

\_aliasIO; IOx00C01\_x0C; IO=0000\_0000h; DataPortWrite=FCH::IO::pci\_intr\_index

Bits	Description
7:5	Reserved.
4:0	<b>Pci2IntrMap</b> . Read-write. Reset: 1Fh. If (FCH::IO::pci_intr_index[7] == 1), then Pci2IntrMap specifies the APIC interrupt number that the corresponding PCI interrupt maps to. If (FCH::IO::pci_intr_index[7] == 0), then Pci2IntrMap specifies the PIC interrupt number that the corresponding PCI interrupt maps to.

**LEGACYIOx000 (FCH::IO::dma\_ch\_0)**

Read-write. Reset: 0000h.

Dma\_Ch 0 register

\_aliasIO; LEGACYIOx000; LEGACYIO=FF00\_0000h

Bits	Description
15:0	<b>dma_ch_0</b> . Read-write. Reset: 0000h. DMA1 Ch0 Base and Current Address

**LEGACYIOx002 (FCH::IO::dma\_ch\_1)**

Read-write. Reset: 0000h.

Dma\_Ch 1 register

\_aliasIO; LEGACYIOx002; LEGACYIO=FF00\_0000h

Bits	Description
15:0	<b>dma_ch_1</b> . Read-write. Reset: 0000h. DMA1 Ch1 Base and Current Address

**LEGACYIOx004 (FCH::IO::dma\_ch\_2)**

Read-write. Reset: 0000h.

Dma\_Ch 2 register

\_aliasIO; LEGACYIOx004; LEGACYIO=FF00\_0000h

Bits	Description
15:0	<b>dma_ch_2</b> . Read-write. Reset: 0000h. DMA2 Ch2 Base and Current Address

**LEGACYIOx006 (FCH::IO::dma\_ch\_3)**

Read-write. Reset: 0000h.

Dma\_Ch 3 register

\_aliasIO; LEGACYIOx006; LEGACYIO=FF00\_0000h

Bits	Description
15:0	<b>dma_ch_3</b> . Read-write. Reset: 0000h. DMA1 Ch3 Base and Current Address

**LEGACYIOx008 (FCH::IO::dma\_status)**

Read-write. Reset: 00h.

Dma\_Status register

\_aliasIO; LEGACYIOx008; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_status</b> . Read-write. Reset: 00h. Returns status when read command for write



**LEGACYIOx009 (FCH::IO::dma\_writerequest)**

Read-write. Reset: 00h.

Dma\_WriteRequest register

\_aliasIO; LEGACYIOx009; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_writerequest.</b> Read-write. Reset: 00h. Request register.

**LEGACYIOx00A (FCH::IO::dma\_writemask)**

Read-write. Reset: 00h.

Dma\_WriteMask register

\_aliasIO; LEGACYIOx00A; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_writemask.</b> Read-write. Reset: 00h. Channel mask register.

**LEGACYIOx00B (FCH::IO::dma\_writemode)**

Read-write. Reset: 00h.

Dma\_WriteMode register

\_aliasIO; LEGACYIOx00B; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_writemode.</b> Read-write. Reset: 00h. Mode register.

**LEGACYIOx00C (FCH::IO::dma\_clear)**

Read-write. Reset: 00h.

Dma\_Clear register

\_aliasIO; LEGACYIOx00C; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_clear.</b> Read-write. Reset: 00h. Channel 0-3 DMA clear byte pointer

**LEGACYIOx00D (FCH::IO::dma\_masterclr)**

Read-write. Reset: 00h.

Dma\_MasterClr register

\_aliasIO; LEGACYIOx00D; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_masterclr.</b> Read-write. Reset: 00h. Intermediate register.

**LEGACYIOx00E (FCH::IO::dma\_clrmask)**

Read-write. Reset: 00h.

Dma\_ClrMask register

\_aliasIO; LEGACYIOx00E; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_clrmask.</b> Read-write. Reset: 00h. Channel 0-3 DMA Clear Mask

**LEGACYIOx00F (FCH::IO::dma\_allmask)**

Read-write. Reset: 00h.

Dma\_AllMask register

\_aliasIO; LEGACYIOx00F; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_allmask.</b> Read-write. Reset: 00h. Mask register.

**LEGACYIOx020 (FCH::IO::intrcntrl1reg1)**

Read-write. Reset: 00h.

IntrCntrl1Reg1 register

\_aliasIO; LEGACYIOx020; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>intrcntrl1reg1</b> . Read-write. Reset: 00h. <b>Description:</b> IRQ0 IRQ7: Read IRR, ISR Write ICW1, OCW2, OCW3

**LEGACYIOx021 (FCH::IO::intrcntrl1reg2)**

Read-write. Reset: 00h.

IntrCntrl1Reg2 register

\_aliasIO; LEGACYIOx021; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>intrcntrl1reg2</b> . Read-write. Reset: 00h. <b>Description:</b> IRQ0 IRQ7: Read IMR Write ICW2, ICW3, ICW4, OCW1

**LEGACYIOx022 (FCH::IO::imcr\_index)**

Read-write. Reset: 00h.

IMCR\_Index register

\_aliasIO; LEGACYIOx022; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>imcr_index</b> . Read-write. Reset: 00h. The IMCR is supported by two read/writeable IO ports 22/23h which are used as index and data port respectively. The actual IMCR register is located at index 70h.

**LEGACYIOx023 (FCH::IO::imcr\_data)**

Read-write. Reset: 00h.

IMCR\_Data register

\_aliasIO; LEGACYIOx023; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>imcr_data</b> . Read-write. Reset: 00h. The IMCR is supported by two read/writeable IO ports 22/23h which are used as index and data port respectively. The actual IMCR register is located at index 70h and it is at bit 0. The actual IMCR bit can only be accessed when bit port 22 is set to 70h. Default value of IMCR is 0.

**LEGACYIOx040 (FCH::IO::timerch0)**

Read-write. Reset: 00h.

TimerCh0 register

\_aliasIO; LEGACYIOx040; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>timerch0</b> . Read-write. Reset: 00h. <b>Description:</b> 8254 Timer 1: Counter 0 Data Port This timer is known as the System Clock timer and it is always on. It is clocked internally by OSC/12 (1.19318MHz), and asserts IRQ0 every time the timer rolls over. This timer is used for time-of-day, diskette time-out, and other system timing functions.

**LEGACYIOx041 (FCH::IO::timerch1)**

Read-write. Reset: 00h.

TimerCh1 register

\_aliasIO; LEGACYIOx041; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>timerch1</b> . Read-write. Reset: 00h.
	<b>Description:</b> 8254 Timer 1: Counter 1 Data Port This timer is normally used for ISA refresh cycles and is also clocked by OSC/12 (1.19818MHz). Since this refresh function is no longer needed (we don't have an external ISA bus), it can be used as a general purpose timing function.

**LEGACYIOx042 (FCH::IO::timerch2)**

Read-write. Reset: 00h.

TimerCh2 register

\_aliasIO; LEGACYIOx042; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>timerch2</b> . Read-write. Reset: 00h.
	<b>Description:</b> 8254 Timer 1: Counter 2 Data Port This is the speaker tone generator and is enabled by IO port 61H. It is clocked by OSC/12 (1.19318MHz) and directly drives the output SPKR that goes to a speaker.

**LEGACYIOx043 (FCH::IO::tmr1cntrlword)**

Write-only. Reset: 00h.

Tmr1CntrlWord register: This is the control word to access the 8254 timer 1. It is used to select which counter will be accessed and how it will be accessed. This register specifies the counter, the operating mode, the order and size of the count value, and whether it counts down in a 16 bit or BCD format.

If a counter is programmed to read or write two-byte counts, the following precaution applies: A program must not transfer control between writing the first and second byte to another routine which also writes into that same counter otherwise, the counter will be loaded with an incorrect value. The count must always be completely loaded with both bytes.

Tmr1CntrlWord is write-only. Read always returns FF.

\_aliasIO; LEGACYIOx043; LEGACYIO=FF00\_0000h

Bits	Description
7:6	<b>counterselect.</b> Write-only. Reset: 0h. <b>Description:</b> 00: Select counter 0 01: Select counter 1 10: Select counter 2 11: Read back command
5:4	<b>cmmandselect.</b> Write-only. Reset: 0h. <b>Description:</b> 00: Counter latch command 01: Read/write least significant byte 10: Read/write most significant byte 11: Read/write least, and then most significant byte
3:1	<b>modeselect.</b> Write-only. Reset: 0h. <b>Description:</b> 000: Asserts OUT signal at end of count 001: Hardware re-triggerable one-shot 010: Rate generator 011: Square wave output 100: Software triggered strobe 101: Hardware triggered strobe 110 111: Not used
0	<b>cntdownselect.</b> Write-only. Reset: 0. <b>Description:</b> 0: Binary countdown 1: BCD countdown

**LEGACYIOx060 (FCH::IO::io\_port\_60)**

Read-write. Reset: 00h.

\_aliasIO; LEGACYIOx060; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>io_port_60.</b> Read-write. Reset: 00h. This is for Legacy USB emulation, please refer to 4.25 USB Legacy register for detail

**LEGACYIOx061 (FCH::IO::nmi\_status)**

Nmi\_Status register: Independent read and write registers will be accessed at this port. When writing to port 61H, bits[3:0] allow software to enable/disable parity error NMI's and control the speaker timer. When reading port 61H, status on parity errors, speaker count, speaker control and refresh cycles is returned.

\_aliasIO; LEGACYIOx061; LEGACYIO=FF00\_0000h

Bits	Description
7	<b>parerr_nmi.</b> Read-only. Reset: X. NMI is caused by parity error (either PERR# or SERR#).
6	<b>iochk_nmi.</b> Read-only. Reset: X. NMI is triggered by serial IOCHK.
5	<b>spkrclk.</b> Read-only. Reset: X. The output of the counter 2.
4	<b>refclk.</b> Read-only. Reset: X. The output of the counter 1 (8254).
3	<b>iochk_nmi_en.</b> Read-write. Reset: 1. <b>Description:</b> 0: Enable IoChk to NMI generation 1: Disable IoChk to NMI generation
2	<b>parity_nmi_en.</b> Read-write. Reset: 1. <b>Description:</b> 0: Enable Parity Error to NMI generation (from SERR# or PERR#) 1: Disable Parity Error to NMI generation and clear bit 7
1	<b>spkrtmrenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Speaker timer off 1: Speaker timer on
0	<b>spkrenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable counter 2 1: Enable counter 2

**LEGACYIOx064 (FCH::IO::io\_port\_64)**

Read-write. Reset: 00h.

\_aliasIO; LEGACYIOx064; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>io_port_64.</b> Read-write. Reset: 00h. This is for Legacy USB emulation, please refer to 4.25 USB Legacy register for detail

**LEGACYIOx070 (FCH::IO::nmi\_enable)**

Read-write. Reset: 80h.

Nmi\_Enable register

\_aliasIO; LEGACYIOx070; LEGACYIO=FF00\_0000h

Bits	Description
7	<b>nmienableb.</b> Read-write. Reset: 1. <b>Description:</b> 0: NMI enable 1: NMI disable
6:0	<b>rtc_address_port.</b> Read-write. Reset: 00h. This is used with either internal RTC or external RTC

**LEGACYIOx071 (FCH::IO::rtcdataport)**

Read-write. Reset: 00h.

RtcDataPort

\_aliasIO; LEGACYIOx071; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>rtc_data_port.</b> Read-write. Reset: 00h. This is used with either internal RTC or external RTC

**LEGACYIOx072 (FCH::IO::alternatrtcaddrport)**

Read-write. Reset: 00h.

AlternatRtcAddrPort

\_aliasIO; LEGACYIOx072; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>alternatrtcaddrport.</b> Read-write. Reset: 00h. This is used with internal RTC. This port allows user to specify the full 8 bit address (instead of bank0/bank1 indexing) to access the 256 byte RTC RAM

**LEGACYIOx073 (FCH::IO::alternatrtcdataport)**

Read-write. Reset: 00h.

AlternatRtcDataPort

\_aliasIO; LEGACYIOx073; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>alternatrtc_data_port.</b> Read-write. Reset: 00h. This is used with internal RTC in conjunction with port h72

**LEGACYIOx080 (FCH::IO::io\_port\_80)**

Read-write. Reset: 00h.

\_aliasIO; LEGACYIOx080; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>io_port_80.</b> Read-write. Reset: 00h. Write to this port will send data to be displayed on "POST Code display module"

**LEGACYIOx081 (FCH::IO::dma\_pagech2)**

Read-write. Reset: 00h.

Dma\_PageCh2 register

\_aliasIO; LEGACYIOx081; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_pagech2.</b> Read-write. Reset: 00h. DMA2 ch 2 page register

**LEGACYIOx082 (FCH::IO::dma\_pagech3)**

Read-write. Reset: 00h.

Dma\_PageCh3 register

\_aliasIO; LEGACYIOx082; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_pagech3.</b> Read-write. Reset: 00h. DMA2 ch 3 page register

**LEGACYIOx083 (FCH::IO::dma\_pagech1)**

Read-write. Reset: 00h.

Dma\_PageCh1 register

\_aliasIO; LEGACYIOx083; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_pagech1.</b> Read-write. Reset: 00h. DMA2 ch 1 page register

**LEGACYIOx084 (FCH::IO::dma\_page\_reserved1)**

Read-write. Reset: 00h.

Dma\_Page\_Reserved1 register

\_aliasIO; LEGACYIOx084; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_page_reserved1.</b> Read-write. Reset: 00h. DMA Page Reserved1 register

**LEGACYIOx085 (FCH::IO::dma\_page\_reserved2)**

Read-write. Reset: 00h.

Dma\_Page\_Reserved2 register

\_aliasIO; LEGACYIOx085; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_page_reserved2.</b> Read-write. Reset: 00h. DMA Page Reserved2 register

**LEGACYIOx086 (FCH::IO::dma\_page\_reserved3)**

Read-write. Reset: 00h.

Dma\_Page\_Reserved3 register

\_aliasIO; LEGACYIOx086; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_page_reserved3.</b> Read-write. Reset: 00h. DMA Page Reserved3 register

**LEGACYIOx087 (FCH::IO::dma\_pagech0)**

Read-write. Reset: 00h.

Dma\_PageCh0 register

\_aliasIO; LEGACYIOx087; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_pagech0.</b> Read-write. Reset: 00h. DMA2 ch 0 page register

**LEGACYIOx088 (FCH::IO::dma\_page\_reserved4)**

Read-write. Reset: 00h.

Dma\_Page\_Reserved4 register

\_aliasIO; LEGACYIOx088; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_page_reserved4.</b> Read-write. Reset: 00h. Dma Page Reserved4 register

**LEGACYIOx089 (FCH::IO::dma\_pagech6)**

Read-write. Reset: 00h.

Dma\_PageCh6 register

\_aliasIO; LEGACYIOx089; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_pagech6.</b> Read-write. Reset: 00h. DMA2 ch 6 page register

**LEGACYIOx08A (FCH::IO::dma\_pagech7)**

Read-write. Reset: 00h.

Dma\_PageCh7 register

\_aliasIO; LEGACYIOx08A; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_pagech7.</b> Read-write. Reset: 00h. DMA2 ch 7 page register

**LEGACYIOx08B (FCH::IO::dma\_pagech5)**

Read-write. Reset: 00h.

Dma\_PageCh5 register

\_aliasIO; LEGACYIOx08B; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_pagech5.</b> Read-write. Reset: 00h. DMA2 ch 5 page register

**LEGACYIOx08C (FCH::IO::dma\_page\_reserved5)**

Read-write. Reset: 00h.

Dma\_Page\_Reserved5 register

\_aliasIO; LEGACYIOx08C; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_page_reserved5.</b> Read-write. Reset: 00h. Dma Page Reserved5 register

**LEGACYIOx08D (FCH::IO::dma\_page\_reserved6)**

Read-write. Reset: 00h.

Dma\_Page\_Reserved6 register

\_aliasIO; LEGACYIOx08D; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_page_reserved6.</b> Read-write. Reset: 00h. Dma Page Reserved6 register

**LEGACYIOx08E (FCH::IO::dma\_page\_reserved7)**

Read-write. Reset: 00h.

Dma\_Page\_Reserved7 register

\_aliasIO; LEGACYIOx08E; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_page_reserved7.</b> Read-write. Reset: 00h. Dma Page Reserved7 register

**LEGACYIOx08F (FCH::IO::dma\_refresh)**

Read-write. Reset: 00h.

Dma\_Refresh register

\_aliasIO; LEGACYIOx08F; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_refresh.</b> Read-write. Reset: 00h. DMA2 ch4 page register.

**LEGACYIOx092 (FCH::IO::fastinit)**

Read-write. Reset: 00h.

\_aliasIO; LEGACYIOx092; LEGACYIO=FF00\_0000h

Bits	Description
7:2	Reserved.
1	<b>a20enb.</b> Read-write. Reset: 0. <b>Description:</b> A20Enable Bar bit if set to 1 A20M# function is disabled.
0	<b>fastinit.</b> Read-write. Reset: 0. <b>Description:</b> FAST_INIT. This read/write bit provides a fast software executed processor reset function. Writing a 1 to this bit will cause the INIT assertion for approximately 4ms. Before another INIT pulse can be generated via this register, this bit must be written back to a 0.

**LEGACYIOx0A0 (FCH::IO::intrcntrl2reg1)**

Read-write. Reset: 00h.

IntrCntrl2Reg1 register

\_aliasIO; LEGACYIOx0A0; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>intrcntrl2reg1.</b> Read-write. Reset: 00h. <b>Description:</b> IRQ8 IRQ15: Read IRR, ISR Write ICW1, OCW2, OCW3



**LEGACYIOx0A1 (FCH::IO::intrcntrl2reg2)**

Read-write. Reset: 00h.

IntrCntrl2Reg2 register

\_aliasIO; LEGACYIOx0A1; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>intrcntrl2reg2.</b> Read-write. Reset: 00h. <b>Description:</b> IRQ8 IRQ15: Read IMR Write ICW2, ICW3, ICW4, OCW1

**LEGACYIOx0C0 (FCH::IO::dma2\_ch4addr)**

Read-write. Reset: 00h.

Dma2\_Ch4Addr register

\_aliasIO; LEGACYIOx0C0; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma2_ch4addr.</b> Read-write. Reset: 00h. DMA2 Ch4 Base and Current Address

**LEGACYIOx0C2 (FCH::IO::dma2\_ch4cnt)**

Read-write. Reset: 00h.

Dma2\_Ch4Cnt register

\_aliasIO; LEGACYIOx0C2; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma2_ch4cnt.</b> Read-write. Reset: 00h. DMA2 Ch4 Base and Current Count

**LEGACYIOx0C4 (FCH::IO::dma2\_ch5addr)**

Read-write. Reset: 00h.

Dma2\_Ch5Addr register

\_aliasIO; LEGACYIOx0C4; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma2_ch5addr.</b> Read-write. Reset: 00h. DMA2 Ch5 Base and Current Address

**LEGACYIOx0C6 (FCH::IO::dma2\_ch5cnt)**

Read-write. Reset: 00h.

Dma2\_Ch5Cnt register

\_aliasIO; LEGACYIOx0C6; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma2_ch5cnt.</b> Read-write. Reset: 00h. DMA2 Ch4 Base and Current Count

**LEGACYIOx0C8 (FCH::IO::dma2\_ch6addr)**

Read-write. Reset: 00h.

Dma2\_Ch6Addr register

\_aliasIO; LEGACYIOx0C8; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma2_ch6addr.</b> Read-write. Reset: 00h. DMA2 Ch6 Base and Current Address

**LEGACYIOx0CA (FCH::IO::dma2\_ch6cnt)**

Read-write. Reset: 00h.

Dma2\_Ch6Cnt register

\_aliasIO; LEGACYIOx0CA; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma2_ch6cnt.</b> Read-write. Reset: 00h. DMA2 Ch6 Base and Current Count

**LEGACYIOx0CC (FCH::IO::dma2\_ch7addr)**

Read-write. Reset: 00h.

Dma2\_Ch7Addr register

\_aliasIO; LEGACYIOx0CC; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma2_ch7addr.</b> Read-write. Reset: 00h. DMA2 Ch5 Base and Current Address

**LEGACYIOx0CE (FCH::IO::dma\_ch7cnt)**

Read-write. Reset: 00h.

Dma2\_Ch7Cnt register

\_aliasIO; LEGACYIOx0CE; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma2_ch7cnt.</b> Read-write. Reset: 00h. Channel 7 DMA base and current count

**LEGACYIOx0D0 (FCH::IO::dma2\_status)**

Read-write. Reset: 00h.

Dma\_Status register

\_aliasIO; LEGACYIOx0D0; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_status.</b> Read-write. Reset: 00h. DMA2 status register

**LEGACYIOx0D2 (FCH::IO::dma2\_writerequest)**

Read-write. Reset: 00h.

Dma\_WriteRequest register

\_aliasIO; LEGACYIOx0D2; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_writerequest.</b> Read-write. Reset: 00h. DMA2 request register

**LEGACYIOx0D4 (FCH::IO::dma2\_writemask)**

Read-write. Reset: 00h.

Dma\_WriteMask register

\_aliasIO; LEGACYIOx0D4; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_writemask.</b> Read-write. Reset: 00h. DMA2 channel mask register

**LEGACYIOx0D6 (FCH::IO::dma2\_writemode)**

Read-write. Reset: 00h.

Dma\_WriteMode register

\_aliasIO; LEGACYIOx0D6; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_writemode.</b> Read-write. Reset: 00h. DMA2 mode register

**LEGACYIOx0D8 (FCH::IO::dma2\_clear)**

Read-write. Reset: 00h.

Dma\_Clear register

\_aliasIO; LEGACYIOx0D8; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>dma_clear.</b> Read-write. Reset: 00h. Channel 4-7 clear byte pointer

**LEGACYIOx0DA (FCH::IO::dma2\_master\_clear)**

Read-write. Reset: 00h.

Dma\_Clear register

\_aliasIO; LEGACYIOx0DA; LEGACYIO=FF00\_0000h

**Bits Description**7:0 **dma\_clear.** Read-write. Reset: 00h. Channel 4-7 DMA master clear**LEGACYIOx0DC (FCH::IO::dma2\_clrmask)**

Read-write. Reset: 00h.

Dma\_ClrMask register

\_aliasIO; LEGACYIOx0DC; LEGACYIO=FF00\_0000h

**Bits Description**7:0 **dma\_clrmask.** Read-write. Reset: 00h. Channel 4-7 DMA Clear Mask**LEGACYIOx0DE (FCH::IO::dma2\_allmask)**

Read-write. Reset: 00h.

Dma\_AllMask register

\_aliasIO; LEGACYIOx0DE; LEGACYIO=FF00\_0000h

**Bits Description**7:0 **dma\_allmask.** Read-write. Reset: 00h. DMA2 mask register**LEGACYIOx0F0 (FCH::IO::ncp\_error)**

Read-write. Reset: 00h.

NCP\_Error register: In addition to the WarmBoot function, writing to this port will assert IGNNE# if FERR# is true. If FERR# is false, then write to this port will not assert IGNNE#.

\_aliasIO; LEGACYIOx0F0; LEGACYIO=FF00\_0000h

**Bits Description**

7 **warmboot.** Read-write. Reset: 0.  
**Description:** Warm or cold boot indicator  
0: Cold  
1: Warm, this bit is set when any value is written to this register

6:0 Reserved.

**LEGACYIOx4D0 (FCH::IO::intrededgecontrol)**

Read-write. Reset: 0000h.

IntrEdgeControl register: This register programs each interrupt to be either edge or level sensitive.

\_aliasIO; LEGACYIOx4D0; LEGACYIO=FF00\_0000h

Bits	Description
15	<b>irq15control.</b> Read-write. Reset: 0. <b>Description:</b> 1: Level 0: Edge
14	<b>irq14control.</b> Read-write. Reset: 0. <b>Description:</b> 1: Level 0: Edge
13	Reserved.
12	<b>irq12control.</b> Read-write. Reset: 0. <b>Description:</b> 1: Level 0: Edge
11	<b>irq11control.</b> Read-write. Reset: 0. <b>Description:</b> 1: Level 0: Edge
10	<b>irq10control.</b> Read-write. Reset: 0. <b>Description:</b> 1: Level 0: Edge
9	<b>irq9control.</b> Read-write. Reset: 0. <b>Description:</b> 1: Level 0: Edge
8	<b>irq8control.</b> Read-write. Reset: 0. (Read Only) Always Edge
7	<b>irq7control.</b> Read-write. Reset: 0. <b>Description:</b> 1: Level 0: Edge
6	<b>irq6control.</b> Read-write. Reset: 0. <b>Description:</b> 1: Level 0: Edge
5	<b>irq5control.</b> Read-write. Reset: 0. <b>Description:</b> 1: Level 0: Edge
4	<b>irq4control.</b> Read-write. Reset: 0. <b>Description:</b> 1: Level 0: Edge
3	<b>irq3control.</b> Read-write. Reset: 0. <b>Description:</b> 1: Level 0: Edge
2	Reserved.
1	<b>irq1control.</b> Read-write. Reset: 0. <b>Description:</b> 1: Level 0: Edge
0	<b>irq0control.</b> Read-write. Reset: 0. <b>Description:</b> 1: Level 0: Edge

**LEGACYIOxC00 (FCH::IO::pci\_intr\_index)**

Read-write. Reset: 00h.

Pci\_Intr\_Index register

\_aliasIO; LEGACYIOxC00; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>pci_intr_index</b> . Read-write. Reset: 00h. PCI interrupt index

**LEGACYIOxC01 (FCH::IO::pci\_intr\_data)**

Read-write. Reset: 1Fh.

Pci\_Intr\_Data register

\_aliasIO; LEGACYIOxC01; LEGACYIO=FF00\_0000h

Bits	Description
7:0	<b>pci_intr_data</b> . Read-write. Reset: 1Fh. PCI redirection register

**LEGACYIOxC14 (FCH::IO::pci\_error)**

Pci\_Error register. This register is enabled/disabled by PM\_Reg:00[20].

\_aliasIO; LEGACYIOxC14; LEGACYIO=FF00\_0000h

Bits	Description
7:4	Reserved.
3	<b>perr_nmi</b> . Read-write. Reset: 1. <b>Description:</b> Enable NMI generation from PERR# 0: Enable 1: Disable
2	<b>serr_nmi</b> . Read-write. Reset: 1. <b>Description:</b> Enable NMI generation from SERR# 0: Enable 1: Disable
1	<b>perr_nmi_status</b> . Read-only. Reset: X. Set to 1 when NMI generation is enabled and PERR# has been asserted due to a PCI data parity error. Cleared by writing a one to port 61h, bit 2.
0	<b>serr_nmi_status</b> . Read-only. Reset: X. Set to 1 when NMI generation is enabled and SERR# has been asserted due to a PCI error. Cleared by writing a one to port 61h, bit 2.

**LEGACYIOxCF9 (FCH::IO::io\_port\_cf9)**

Read-write. Reset: 00h.

\_aliasIO; LEGACYIOxCF9; LEGACYIO=FF00\_0000h

Bits	Description
7:4	Reserved.
3:1	<b>io_port_cf9</b> . Read-write. Reset: 0h. <b>Description:</b> 010: do init, send INIT HT message 011: do reset, generate Warm reset 111: do full reset, generate Cold reset and place system in S5 state for 3 to 5 seconds All other values are reserved
0	Reserved.

**RTCHOSTx00000000 (FCH::IO::seconds)**

Read-write.

Seconds register

\_aliasHOST; RTCHOSTx00000000; RTCHOST=FED8\_0700h

Bits	Description
7:0	<b>seconds.</b> Read-write. Reset: XXXXXXXXb.
	<b>Description:</b> Binary-Code-Decimal format. Range:00 59 This register can be set by software (set bit of Register B = 1) or can be automatically updated by hardware every second. When set by software, hardware updating is disabled. Note: the value of this register is undefined/indeterministic when power up first time.

**RTCHOSTx00000001 (FCH::IO::seconds\_alarm)**

Read-write.

Seconds Alarm register

\_aliasHOST; RTCHOSTx00000001; RTCHOST=FED8\_0700h

Bits	Description
7:0	<b>seconds_alarm.</b> Read-write. Reset: XXXXXXXXb.
	<b>Description:</b> Binary-Code-Decimal format. If set bit = 1, the Seconds Alarm Register will never match with Seconds Register, else If bits [7:6] = [11], the Seconds Alarm Register always matches with Seconds Register. Note: the value of this register is undefined/indeterministic when power up first time.

**RTCHOSTx00000002 (FCH::IO::minutes)**

Read-write.

Minutes register

\_aliasHOST; RTCHOSTx00000002; RTCHOST=FED8\_0700h

Bits	Description
7:0	<b>minutes.</b> Read-write. Reset: XXXXXXXXb.
	<b>Description:</b> Binary-Code-Decimal format. Range:00 59 This register can be set by software (set bit of Register B = 1) or can be automatically updated by hardware every minute. When set by software, hardware updating is disabled. Note: the value of this register is undefined/indeterministic when power up first time.

**RTCHOSTx00000003 (FCH::IO::minutes\_alarm)**

Read-write.

Minutes Alarm register

\_aliasHOST; RTCHOSTx00000003; RTCHOST=FED8\_0700h

Bits	Description
7:0	<b>minutes_alarm.</b> Read-write. Reset: XXXXXXXXb.
	<b>Description:</b> Binary-Code-Decimal format. If set bit = 1, the Minutes Alarm Register will never match with Minutes Register, else If bits [7:6] = [11], the Minutes Alarm Register always matches with Minutes Register. Note: the value of this register is undefined/indeterministic when power up first time.

**RTCHOSTx00000004 (FCH::IO::hours)**

Read-write.

Hours register

\_aliasHOST; RTCHOSTx00000004; RTCHOST=FED8\_0700h

Bits	Description
7:0	<b>hours.</b> Read-write. Reset: XXXXXXXXb.
	<b>Description:</b> Binary-Code-Decimal format. Range:00 23 This register can be set by software (set bit of Register B = 1) or can be automatically updated by hardware every hour. When set by software, hardware updating is disabled. Note: the value of this register is undefined/indeterministic when power up first time.

**RTCHOSTx00000005 (FCH::IO::hours\_alarm)**

Read-write.

Hours Alarm register

\_aliasHOST; RTCHOSTx00000005; RTCHOST=FED8\_0700h

Bits	Description
7:0	<b>hours_alarm.</b> Read-write. Reset: XXXXXXXXb.
	<b>Description:</b> Binary-Code-Decimal format. If set bit = 1, the Hours Alarm Register will never match with Hours Register, else If bits [7:6] = [11], the Hours Alarm Register always matches with Hours Register. Note: the value of this register is undefined/indeterministic when power up first time.

**RTCHOSTx00000006 (FCH::IO::day\_of\_week)**

Read-write.

Day of Week register

\_aliasHOST; RTCHOSTx00000006; RTCHOST=FED8\_0700h

Bits	Description
7:0	<b>day_of_week.</b> Read-write. Reset: XXXXXXXXb.
	<b>Description:</b> Binary-Code-Decimal format. Range: 01 07 (Sunday = 1). No leap year correction capability. Leap year correction has to be done by software. This register can be set by a software (set bit of Register B = 1) or can be automatically updated by hardware everyday. When set by software, hardware updating is disabled. Note: the value of this register is undefined/indeterministic when power up first time.

**RTCHOSTx00000007 (FCH::IO::date\_of\_month)**

Read-write.

Date of Month register

\_aliasHOST; RTCHOSTx00000007; RTCHOST=FED8\_0700h

Bits	Description
7:0	<b>date_of_month.</b> Read-write. Reset: XXXXXXXXb.
	<b>Description:</b> Binary-Code-Decimal format. Range: 01 28 for February and no leap year capability. Leap year correction has to be done by software. This register can be set by software (set bit of Register B = 1) or can be automatically updated by hardware everyday. When set by software, hardware updating is disabled. Note: the value of this register is undefined/indeterministic when power up first time.

**RTCHOSTx00000008 (FCH::IO::month)**

Read-write.

Month register

\_aliasHOST; RTCHOSTx000000008; RTCHOST=FED8\_0700h

Bits	Description
7:0	<b>month.</b> Read-write. Reset: XXXXXXXXb. <b>Description:</b> Binary-Code-Decimal format. Range: 01 12. No leap year correction capability. Leap year correction has to be done by software. This register can be set by software (set bit of Register B = 1) or can be automatically updated by hardware every month. When set by software, hardware updating is disabled. Note: the value of this register is undefined/indeterministic when power up first time.

**RTCHOSTx00000009 (FCH::IO::year)**

Read-write.

Year register

\_aliasHOST; RTCHOSTx000000009; RTCHOST=FED8\_0700h

Bits	Description
7:0	<b>year.</b> Read-write. Reset: XXXXXXXXb. <b>Description:</b> Binary-Code-Decimal format. Range: 00 99. No leap year correction capability. Leap year correction has to be done by software. This register can be set by software (set bit of Register B = 1) or can be automatically updated by hardware every year. When set by software, hardware updating is disabled. Note: the value of this register is undefined/indeterministic when power up first time.

**RTCHOSTx0000000A (FCH::IO::register\_a)**

Read-write.

Register A: Control register

\_aliasHOST; RTCHOSTx00000000A; RTCHOST=FED8\_0700h

Bits	Description
7	<b>update_in_progress_uip.</b> Read-write. Reset: X. <b>Description:</b> If set bit = 1, UIP is cleared. If UIP = 1, the update transfer will soon occur. If UIP = 0, the update transfer will not occur for at least 244us. [Read-only] Note: the value of this register is undefined/indeterministic when power up first time.
6:5	Reserved.
4	<b>bank_selection_dv0.</b> Read-write. Reset: X. <b>Description:</b> DV0 = 0 selects Bank 0 DV0 = 1 selects Bank 1. The SB800 has an alternate way to access the RAM without the use of bank select bit. Port 72/73 can be used as the index to access the full 256 bytes of RAM directly. Note: the value of this register is undefined/indeterministic when power up first time.
3	<b>rate_selection_rs3.</b> Read-write. Reset: X.
2	<b>rate_selection_rs2.</b> Read-write. Reset: X.
1	<b>rate_selection_rs1.</b> Read-write. Reset: X.
0	<b>rate_selection_rs0.</b> Read-write. Reset: X. <b>Description:</b> These four rate-selection bits select one of the 13 taps on the 15-stage frequency divider or disable the divider output (flat output signal). The tap selected can be used to generate a periodic interrupt. See the following table for the frequency selection. Note: the value of this register is undefined/indeterministic when power up first time.



**RTCHOSTx0000000B (FCH::IO::register\_b)**

Read-write.

Register B: Control register

\_aliasHOST; RTCHOSTx0000000B; RTCHOST=FED8\_0700h

Bits	Description
7	<b>set_new_time_set.</b> Read-write. Reset: X. <b>Description:</b> If set bit = 1, no internal updating for Time Registers is allowed. If set bit = 0, the Time Registers are updated every second. Note: the value of this register is undefined/indeterministic when power up first time.
6	<b>periodic_interrupt_enable_pie.</b> Read-write. Reset: X. <b>Description:</b> PIE enables the Periodic Interrupt Flag (PF) bit in Register C to assert IRQ. Note: the value of this register is undefined/indeterministic when power up first time.
5	<b>alarm_interrupt_enable_aie.</b> Read-write. Reset: X. <b>Description:</b> AIE enables the Alarm Flag (AF) bit in Register C to assert IRQ. Note: the value of this register is undefined/indeterministic when power up first time.
4	<b>update_ended_interrupt_enable_uie.</b> Read-write. Reset: X. <b>Description:</b> UIE enables the Update End Flag (UF) bit in Register C to assert IRQ. If set bit = 1, UIE is cleared. Note: the value of this register is undefined/indeterministic when power up first time.
3:2	Reserved.
1	<b>hourmode.</b> Read-write. Reset: X. <b>Description:</b> Hour mode 0: 12 hour mode 1: 24 hour mode Note: the value of this register is undefined/indeterministic when power up first time.
0	<b>daylight_saving_enable.</b> Read-write. Reset: X. <b>Description:</b> Both this bit and RtcExt_Reg: 00h bit[0] need to be set to 1 to enable RTC daylight saving feature. Note: the value of this register is undefined/indeterministic when power up first time.

**RTCHOSTx0000000C (FCH::IO::register\_c)**

Read-only.

Register C: Control register

\_aliasHOST; RTCHOSTx0000000C; RTCHOST=FED8\_0700h

Bits	Description
7	<b>interrupt_request_flag_irqf.</b> Read-only. Reset: X. <b>Description:</b> Logically, IRQF = (PF*PIE)+(AF*AIE)+(UF*UIE)+(WF*WIE) where WF and WIE are defined in Extended Control Register 4A and 4B. Reading Register C clears IRQF bit. Any time the IRQF bit is set to one, the #IRQ pin is driven low. Note: the value of this register is undefined/indeterministic when power up first time.
6	<b>xperiodic_interrupt_flag_pf.</b> Read-only. Reset: X. <b>Description:</b> This bit is set to one when an edge is detected on the selected tap (through RS3 to RS0) of the frequency divider. Reading Register C clears PF bit. Note: the value of this register is undefined/indeterministic when power up first time.
5	<b>alarm_interrupt_flag_af.</b> Read-only. Reset: X. <b>Description:</b> This bit is set to one if second, minute and hour time has matched the second, minute and hour alarm time. Reading Register C clears AF bit. Note: the value of this register is undefined/indeterministic when power up first time.
4	<b>update_ended_interrupt_flag_uf.</b> Read-only. Reset: X. <b>Description:</b> This bit is set to one after each update cycle. Reading Register C clears UF. Note: the value of this register is undefined/indeterministic when power up first time.
3:0	Reserved.

**RTCHOSTx0000000D (FCH::IO::datealarm)**

Read-write.

Date Alarm Register

\_aliasHOST; RTCHOSTx0000000D; RTCHOST=FED8\_0700h

Bits	Description
7	<b>vrt.</b> Read-write. Reset: X. <b>Description:</b> Valid RAM and Time refer to VRT_T1 and VRT_T2 registers (PMIO 3E/3F) Note: the value of this register is undefined/indeterministic when power up first time.
6	<b>scratchbit.</b> Read-write. Reset: X.
5:0	<b>datealarm.</b> Read-write. Reset: XXXXXXb. <b>Description:</b> DateAlarm in BCD format and is considered when it is set to non-zero value. If this value is set to 0, then date is not compared for alarm generation. Note: the value of this register is undefined/indeterministic when power up first time.

**RTCHOSTx00000032 (FCH::IO::altcentury)**

Read-write.

AltCentury Register

\_aliasHOST; RTCHOSTx00000032; RTCHOST=FED8\_0700h

Bits	Description
7:0	<b>altcentury.</b> Read-write. Reset: XXXXXXXXb. <b>Description:</b> (This register is accessed only when DV0=0 and PM_Reg 56h Bit12=1.) Binary-Code-Decimal format. Leap year correction is done through hardware. This register can be set by software (set bit of Register B = 1) or can be automatically updated by hardware every century. When set by software, hardware updating is disabled. Note: the value of this register is undefined/indeterministic when power up first time.

**RTCHOSTx00000048 (FCH::IO::century)**

Read-write.

Century Register

\_aliasHOST; RTCHOSTx00000048; RTCHOST=FED8\_0700h

Bits	Description
7:0	<b>century.</b> Read-write. Reset: XXXXXXXXb. <b>Description:</b> (This register is accessed only when DV0=1) Binary-Code-Decimal format. Leap year correction is done through hardware. This register can be set by software (set bit of Register B = 1) or can be automatically updated by hardware every century. When set by software, hardware updating is disabled. Note: the value of this register is undefined/indeterministic when power up first time.

**RTCHOSTx00000050 (FCH::IO::extended\_ram\_address\_port)**

Read-write.

Extended RAM Address Port register: The address port to access Extended RAM.

\_aliasHOST; RTCHOSTx00000050; RTCHOST=FED8\_0700h

Bits	Description
7	Reserved.
6:0	<b>extendedramaddr.</b> Read-write. Reset: XXXXXXXb. <b>Description:</b> Because only 7 address bits are used in port x70, only lower 128 bytes are accessible through port x71. The Extended RAM (upper 128 bytes) are physically located at address 80H to FFH. In order to access these address, an address offset should be programmed into this register and access them through Extended RAMDataPort. (An offset of x80H will automatically add to this 7-bit address). Note: the value of this register is undefined/indeterministic when power up first time.

**RTCHOSTx00000053 (FCH::IO::extended\_ram\_data\_port)**

Read-write.

Extended RAM Data Port register.

\_aliasHOST; RTCHOSTx00000053; RTCHOST=FED8\_0700h

Bits	Description
7:0	<b>extended_ram_data_port</b> . Read-write. Reset: XXXXXXXXb. There is no physical register corresponding to this data port but the data port address is used for decoding to generate appropriate internal control signals.

**RTCHOSTx0000007E (FCH::IO::rtc\_time\_clear)**

Read-write.

RTC Time Clear register.

\_aliasHOST; RTCHOSTx0000007E; RTCHOST=FED8\_0700h

Bits	Description
7:1	Reserved.
0	<b>rtctimeclear</b> . Read-write. Reset: X. <b>Description:</b> Setting this bit '1' will clear the RTC second and RTC time will stop. When PSP_regxFC[9]=1, this bit can only be written '0'. Note: the value of this register is undefined/indeterministic when power up first time.

**RTCHOSTx0000007F (FCH::IO::rtc\_ram\_enable)**

Read-write.

RTC RAM Enable register.

\_aliasHOST; RTCHOSTx0000007F; RTCHOST=FED8\_0700h

Bits	Description
7:1	Reserved.
0	<b>rtcramenable</b> . Read-write. Reset: X. <b>Description:</b> Setting this bit will enable access to the RTC RAM Note: the value of this register is undefined/indeterministic when power up first time.

**7.3.2 I/O Advanced Programmable Interrupt Control****7.3.2.1 IOAPIC Registers****IOAPICx00000000 (FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER)**

Read-write.

Used to determine which register is manipulated during an IO Window Register read/write operation.

\_aliasHOST; IOAPICx00000000; IOAPIC=FEC0\_0000h

Bits	Description
31:8	Reserved.
7:0	<b>indirect_address_offset</b> . Read-write. Reset: 00h. Indirect Address Offset to IO Window Register, used to determine which register is manipulated during an IO Window Register read/write operation.

**IOAPICx00000010 (FCH::IOAPIC::IO\_WINDOW\_REGISTER)**

Read-write. Reset: 0000\_0000h.

`_aliasHOST; IOAPICx00000010; IOAPIC=FEC0_0000h`

Bits	Description
31:0	<b>io_window.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> Mapped by the value in the IO Register Select Register to the designated indirect access register. Technically a RW register however, the read/write capability is determined by the indirect access register referenced by the IO Register Select Register.

**IOAPICx00000010\_indirectaddressoffset00 (FCH::IOAPIC::ioapic\_id\_register)**

Read-write. Reset: 0000\_0000h.

Not used in XAPIC PCI bus delivery mode.

`_aliasHOST; IOAPICx00000010_indirectaddressoffset00; IOAPIC=FEC0_0000h;  
DataPortWrite=FCH::IOAPIC::IO_REGISTER_SELECT_REGISTER[indirect_address_offset]`

Bits	Description
31:24	<b>id.</b> Read-write. Reset: 00h. IOAPIC device ID for APIC serial bus delivery mode
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset01 (FCH::IOAPIC::ioxapic\_version\_register)**

Read-only. Reset: 0017\_8021h.

`_aliasHOST; IOAPICx00000010_indirectaddressoffset01; IOAPIC=FEC0_0000h;  
DataPortWrite=FCH::IOAPIC::IO_REGISTER_SELECT_REGISTER[indirect_address_offset]`

Bits	Description
31:24	Reserved.
23:16	<b>max_redirection_entries.</b> Read-only. Reset: 17h. 24 entries [23:0]
15	<b>prq.</b> Read-only. Reset: 1. IRQ pin assertion supported
14:8	Reserved.
7:0	<b>version.</b> Read-only. Reset: 21h. PCI 2.2 compliant

**IOAPICx00000010\_indirectaddressoffset02 (FCH::IOAPIC::ioapic\_arbitration\_register)**

Read-only. Reset: 0000\_0000h.

Not used in XAPIC PCI bus delivery mode.

`_aliasHOST; IOAPICx00000010_indirectaddressoffset02; IOAPIC=FEC0_0000h;  
DataPortWrite=FCH::IOAPIC::IO_REGISTER_SELECT_REGISTER[indirect_address_offset]`

Bits	Description
31:28	Reserved.
27:24	<b>arbitration_id.</b> Read-only. Reset: 0h. Arbitration ID for APIC serial bus delivery mode
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset10 (FCH::IOAPIC::redirection\_table\_entry\_0\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset10; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask.</b> Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr.</b> Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity.</b> Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status.</b> Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode.</b> Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector.</b> Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset11 (FCH::IOAPIC::redirection\_table\_entry\_0\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset11; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id.</b> Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset12 (FCH::IOAPIC::redirection\_table\_entry\_1\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset12; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask.</b> Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr.</b> Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity.</b> Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status.</b> Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode.</b> Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector.</b> Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset13 (FCH::IOAPIC::redirection\_table\_entry\_1\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset13; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id.</b> Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset14 (FCH::IOAPIC::redirection\_table\_entry\_2\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset14; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask.</b> Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr.</b> Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity.</b> Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status.</b> Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode.</b> Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector.</b> Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset15 (FCH::IOAPIC::redirection\_table\_entry\_2\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset15; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id.</b> Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset16 (FCH::IOAPIC::redirection\_table\_entry\_3\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset16; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask.</b> Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr.</b> Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity.</b> Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status.</b> Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode.</b> Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector.</b> Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset17 (FCH::IOAPIC::redirection\_table\_entry\_3\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset17; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id.</b> Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.



**IOAPICx00000010\_indirectaddressoffset18 (FCH::IOAPIC::redirection\_table\_entry\_4\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset18; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask.</b> Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr.</b> Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity.</b> Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status.</b> Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode.</b> Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector.</b> Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset19 (FCH::IOAPIC::redirection\_table\_entry\_4\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset19; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id.</b> Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset1A (FCH::IOAPIC::redirection\_table\_entry\_5\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset1A; IOAPIC=FEC0\_0000h;

DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask</b> . Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr</b> . Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity</b> . Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status</b> . Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode</b> . Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector</b> . Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset1B (FCH::IOAPIC::redirection\_table\_entry\_5\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset1B; IOAPIC=FEC0\_0000h;

DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id</b> . Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset1C (FCH::IOAPIC::redirection\_table\_entry\_6\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset1C; IOAPIC=FEC0\_0000h;

DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask.</b> Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr.</b> Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity.</b> Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status.</b> Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode.</b> Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector.</b> Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset1D (FCH::IOAPIC::redirection\_table\_entry\_6\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset1D; IOAPIC=FEC0\_0000h;

DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id.</b> Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset1E (FCH::IOAPIC::redirection\_table\_entry\_7\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset1E; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask.</b> Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr.</b> Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity.</b> Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status.</b> Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode.</b> Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector.</b> Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset1F (FCH::IOAPIC::redirection\_table\_entry\_7\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset1F; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id.</b> Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset20 (FCH::IOAPIC::redirection\_table\_entry\_8\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset20; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask</b> . Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr</b> . Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity</b> . Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status</b> . Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode</b> . Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector</b> . Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset21 (FCH::IOAPIC::redirection\_table\_entry\_8\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset21; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id</b> . Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset22 (FCH::IOAPIC::redirection\_table\_entry\_9\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset22; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask</b> . Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr</b> . Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity</b> . Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status</b> . Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode</b> . Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector</b> . Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset23 (FCH::IOAPIC::redirection\_table\_entry\_9\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset23; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id</b> . Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset24 (FCH::IOAPIC::redirection\_table\_entry\_10\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset24; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask.</b> Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr.</b> Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity.</b> Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status.</b> Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode.</b> Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector.</b> Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset25 (FCH::IOAPIC::redirection\_table\_entry\_10\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset25; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id.</b> Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset26 (FCH::IOAPIC::redirection\_table\_entry\_11\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset26; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask.</b> Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr.</b> Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity.</b> Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status.</b> Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode.</b> Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector.</b> Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset27 (FCH::IOAPIC::redirection\_table\_entry\_11\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset27; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id.</b> Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.



**IOAPICx00000010\_indirectaddressoffset28 (FCH::IOAPIC::redirection\_table\_entry\_12\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset28; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask.</b> Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr.</b> Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity.</b> Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status.</b> Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode.</b> Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector.</b> Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset29 (FCH::IOAPIC::redirection\_table\_entry\_12\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset29; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id.</b> Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset2A (FCH::IOAPIC::redirection\_table\_entry\_13\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset2A; IOAPIC=FEC0\_0000h;

DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask</b> . Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr</b> . Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity</b> . Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status</b> . Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode</b> . Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector</b> . Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset2B (FCH::IOAPIC::redirection\_table\_entry\_13\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset2B; IOAPIC=FEC0\_0000h;

DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id</b> . Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset2C (FCH::IOAPIC::redirection\_table\_entry\_14\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset2C; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask.</b> Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr.</b> Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity.</b> Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status.</b> Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode.</b> Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector.</b> Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset2D (FCH::IOAPIC::redirection\_table\_entry\_14\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset2D; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id.</b> Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset2E (FCH::IOAPIC::redirection\_table\_entry\_15\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset2E; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask</b> . Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr</b> . Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity</b> . Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status</b> . Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode</b> . Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector</b> . Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset2F (FCH::IOAPIC::redirection\_table\_entry\_15\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset2F; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id</b> . Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset30 (FCH::IOAPIC::redirection\_table\_entry\_16\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset30; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask</b> . Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr</b> . Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity</b> . Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status</b> . Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode</b> . Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector</b> . Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset31 (FCH::IOAPIC::redirection\_table\_entry\_16\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset31; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id</b> . Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset32 (FCH::IOAPIC::redirection\_table\_entry\_17\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset32; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask.</b> Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr.</b> Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity.</b> Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status.</b> Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode.</b> Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector.</b> Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset33 (FCH::IOAPIC::redirection\_table\_entry\_17\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset33; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id.</b> Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset34 (FCH::IOAPIC::redirection\_table\_entry\_18\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset34; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask</b> . Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr</b> . Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity</b> . Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status</b> . Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode</b> . Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector</b> . Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset35 (FCH::IOAPIC::redirection\_table\_entry\_18\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset35; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id</b> . Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset36 (FCH::IOAPIC::redirection\_table\_entry\_19\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset36; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask</b> . Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr</b> . Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity</b> . Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status</b> . Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode</b> . Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector</b> . Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset37 (FCH::IOAPIC::redirection\_table\_entry\_19\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset37; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id</b> . Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.



**IOAPICx00000010\_indirectaddressoffset38 (FCH::IOAPIC::redirection\_table\_entry\_20\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset38; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask.</b> Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr.</b> Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity.</b> Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status.</b> Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode.</b> Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector.</b> Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset39 (FCH::IOAPIC::redirection\_table\_entry\_20\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset39; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id.</b> Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset3A (FCH::IOAPIC::redirection\_table\_entry\_21\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset3A; IOAPIC=FEC0\_0000h;

DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask</b> . Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr</b> . Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity</b> . Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status</b> . Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode</b> . Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector</b> . Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset3B (FCH::IOAPIC::redirection\_table\_entry\_21\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset3B; IOAPIC=FEC0\_0000h;

DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id</b> . Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset3C (FCH::IOAPIC::redirection\_table\_entry\_22\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset3C; IOAPIC=FEC0\_0000h;

DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask.</b> Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr.</b> Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity.</b> Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status.</b> Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode.</b> Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector.</b> Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset3D (FCH::IOAPIC::redirection\_table\_entry\_22\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset3D; IOAPIC=FEC0\_0000h;

DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id.</b> Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000010\_indirectaddressoffset3E (FCH::IOAPIC::redirection\_table\_entry\_23\_low\_32bit)**

Read-write. Reset: 0001\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset3E; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:17	Reserved.
16	<b>mask</b> . Read-write. Reset: 1. <b>Description:</b> Masks the interrupt injection at the input of this device. Write 0 to unmask
15	<b>trigger_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Edge 1: Level
14	<b>remote_irr</b> . Read-write. Reset: 0. <b>Description:</b> Read Only. Used for level triggered interrupts only. Set when interrupt message delivered. Cleared by EOI special cycle transaction or write to EOI register
13	<b>interrupt_pin_polarity</b> . Read-write. Reset: 0. <b>Description:</b> 0: High 1: Low
12	<b>delivery_status</b> . Read-write. Reset: 0. <b>Description:</b> Read Only 0: Idle 1: Send Pending
11	<b>destination_mode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Physical 1: Logical
10:8	<b>delivery_mode</b> . Read-write. Reset: 0h. <b>Description:</b> 000: Fixed 001: Lowest Priority 010: SMI/PMI 011: Reserved 100: NMI 101: INIT 110: Reserved 111: ExtINT
7:0	<b>vector</b> . Read-write. Reset: 00h. Interrupt vector associated with this interrupt input

**IOAPICx00000010\_indirectaddressoffset3F (FCH::IOAPIC::redirection\_table\_entry\_23\_high\_32bit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; IOAPICx00000010\_indirectaddressoffset3F; IOAPIC=FEC0\_0000h;  
DataPortWrite=FCH::IOAPIC::IO\_REGISTER\_SELECT\_REGISTER[indirect\_address\_offset]

Bits	Description
31:24	<b>destination_id</b> . Read-write. Reset: 00h. Bits [19:12] of the address field of the interrupt message
23:0	Reserved.

**IOAPICx00000020 (FCH::IOAPIC::irq\_pin\_assertion\_register)**

Read-write. Reset: 0000\_0000h.

Write to this register will trigger an interrupt associated with the redirection table entry referenced by the IRQ number. Currently the redirection table has 24 entries. Write with IRQ number greater than 17H has no effect.

\_aliasHOST; IOAPICx00000020; IOAPIC=FEC0\_0000h

Bits	Description
31:8	Reserved.
7:0	<b>input_irq</b> . Read-write. Reset: 00h. IRQ number for the requested interrupt

**IOAPICx00000040 (FCH::IOAPIC::eoi\_register)**

Write-only. Reset: 0000\_0000h.

Write to this register will clear the remote IRR bit in the redirection table entry found matching the interrupt vector. This provides an alternate mechanism other than PCI special cycle for EOI to reach IOXAPIC.

\_aliasHOST; IOAPICx00000040; IOAPIC=FEC0\_0000h

Bits	Description
31:8	Reserved.
7:0	<b>vector</b> . Write-only. Reset: 00h. Interrupt vector

**7.3.3 SMI Registers**

SMI register space is accessed through the AcpiMmio region. The SMI registers range from FED8\_0000h+200h to FED8\_0000h+2FFh. See FCH::PM::isacontrol[mmioen].

**SMIx00000000 (FCH::SMI::event\_status)**

Read-write.

\_aliasHOST; SMIx00000000; SMI=FED8\_0200h

Bits	Description
31:0	<b>eventstatus</b> . Read-write. Reset: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXb. <b>Description:</b> This is a mirror register of the standard ACPI EVENT_STATUS register. Writing 1 to each bit clears the corresponding status bit. Each Event status is set when the selected event input equals to the corresponding value in SciTrig.

**SMIx00000004 (FCH::SMI::event\_enable)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000004; SMI=FED8\_0200h

Bits	Description
31:0	<b>eventenable</b> . Read-write. Reset: 0000_0000h. This is the mirror register of the standard ACPI EVENT_ENABLE register. Each bit controls whether ACP should generate wakeup and Sci interrupt.

**SMIx00000008 (FCH::SMI::scitrig)**

Read-write. Reset: FFFF\_FFFFh.

\_aliasHOST; SMIx000000008; SMI=FED8\_0200h

Bits	Description
31	<b>scitrig31.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 31 0: Active low 1: Active high
30	<b>scitrig30.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 30 0: Active low 1: Active high
29	<b>scitrig29.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 29 0: Active low 1: Active high
28	<b>scitrig28.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 28 0: Active low 1: Active high
27	<b>scitrig27.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 27 0: Active low 1: Active high
26	<b>scitrig26.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 26 0: Active low 1: Active high
25	<b>scitrig25.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 25 0: Active low 1: Active high
24	<b>scitrig24.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 24 0: Active low 1: Active high
23	<b>scitrig23.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 23 0: Active low 1: Active high
22	<b>scitrig22.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 22 0: Active low 1: Active high
21	<b>scitrig21.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 21 0: Active low 1: Active high
20	<b>scitrig20.</b> Read-write. Reset: 1.

	<b>Description:</b> The bit controls the way to set Event_Status bit 20 0: Active low 1: Active high
19	<b>scitrig19.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 19 0: Active low 1: Active high
18	<b>scitrig18.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 18 0: Active low 1: Active high
17	<b>scitrig17.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 17 0: Active low 1: Active high
16	<b>scitrig16.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 16 0: Active low 1: Active high
15	<b>scitrig15.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 15 0: Active low 1: Active high
14	<b>scitrig14.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 14 0: Active low 1: Active high
13	<b>scitrig13.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 13 0: Active low 1: Active high
12	<b>scitrig12.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 12 0: Active low 1: Active high
11	<b>scitrig11.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 11 0: Falling edge 1: Active high
10	<b>scitrig10.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 10 0: Active low 1: Active high
9	<b>scitrig9.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 9 0: Active low 1: Active high
8	<b>scitrig8.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 8 0: Active low 1: Active high

7	<b>scitrig7.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 7 0: Active low 1: Active high
6	<b>scitrig6.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 6 0: Active low 1: Active high
5	<b>scitrig5.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 5 0: Active low 1: Active high
4	<b>scitrig4.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 4 0: Active low 1: Active high
3	<b>scitrig3.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 3 0: Active low 1: Active high
2	<b>scitrig2.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 2 0: Active low 1: Active high
1	<b>scitrig1.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 1 0: Active low 1: Active high
0	<b>scitrig0.</b> Read-write. Reset: 1. <b>Description:</b> The bit controls the way to set Event_Status bit 0 0: Active low 1: Active high



SMIx0000000C (FCH::SMI::scilevl)	
Read-write. Reset: 0000_0000h.	
_aliasHOST; SMIx0000000C; SMI=FED8_0200h	
Bits	Description
31	<b>scilevl31.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
30	<b>scilevl30.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
29	<b>scilevl29.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
28	<b>scilevl28.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
27	<b>scilevl27.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
26	<b>scilevl26.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
25	<b>scilevl25.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
24	<b>scilevl24.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
23	<b>scilevl23.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
22	<b>scilevl22.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
21	<b>scilevl21.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
20	<b>scilevl20.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
19	<b>scilevl19.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
18	<b>scilevl18.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
17	<b>scilevl17.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
16	<b>scilevl16.</b> Read-write. Reset: 0.

	<b>Description:</b> 0: Edge trigger 1: Level trigger
15	<b>scilevl15.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
14	<b>scilevl14.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
13	<b>scilevl13.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
12	<b>scilevl12.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
11	<b>scilevl11.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
10	<b>scilevl10.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
9	<b>scilevl9.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
8	<b>scilevl8.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>scilevl7.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
6	<b>scilevl6.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
5	<b>scilevl5.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
4	<b>scilevl4.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
3	<b>scilevl3.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
2	<b>scilevl2.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
1	<b>scilevl1.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
0	<b>scilevl0.</b> Read-write. Reset: 0.

	<b>Description:</b> This register defines the trigger mode for each of the Event_Status: 0: Edge trigger 1: Level trigger
--	---

#### SMIx00000010 (FCH::SMI::smiscistatus)

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000010; SMI=FED8\_0200h

Bits	Description
31:0	<b>smiscistatus.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> Each bit indicates the corresponding SmiSci status. The input of each bit is controlled by the corresponding SciTrig bit. Each status bit can be cleared to 0 by writing 1. Note this function can be considered as a superset of Event_Status. When one of this bit is set (and its SmiSciEn is also set), it will trigger a SMI to call the BIOS. After the BIOS has serviced the SMM and clears its status, the internal logic will automatically set the corresponding Event_Status bit and thereby triggering a SCI.

#### SMIx00000014 (FCH::SMI::smiscien)

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000014; SMI=FED8\_0200h

Bits	Description
31:0	<b>smiscien.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> Each bit controls if SMI message will be generated when the corresponding SmiSciStatus bit is set to 1. 0: Not to send SMI message when the corresponding SmiSciStatus bit is set 1: Send SMI message when the corresponding SmiSciStatus bit is set

#### SMIx00000018 (FCH::SMI::softwarescien)

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000018; SMI=FED8\_0200h

Bits	Description
31:0	<b>softwarescien.</b> Read-write. Reset: 0000_0000h. When set, software can write to softwareSciData and set the corresponding Event_Status bit (note the setting of this bit will need to match with SciTrig and SciLevl in order to set the status bit). This register is meant as a software mechanism to trigger SCI.

#### SMIx0000001C (FCH::SMI::softwarescidata)

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx0000001C; SMI=FED8\_0200h

Bits	Description
31:0	<b>softwarescidata.</b> Read-write. Reset: 0000_0000h. This is the software data path to set the corresponding Event_Status when softwareSciEn is set

#### SMIx00000020 (FCH::SMI::scisleepdisable)

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000020; SMI=FED8\_0200h

Bits	Description
31:0	<b>scisleepdisable.</b> Read-write. Reset: 0000_0000h. When set, the corresponding Event_Status bit will be masked off whenever the system goes to S3 or higher sleep state. This is meant for ignoring EVENT pins that are powered in the main power domain (instead of aux. power domain).

**SMIx00000030 (FCH::SMI::captureddata)**

Read-only.

\_aliasHOST; SMIx00000030; SMI=FED8\_0200h

Bits	Description
31:0	<b>captureddata</b> . Read-only. Reset: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXb. This is the buffer to capture write data for the last transaction that caused an SMI#. Note: this buffer has no meaning for a read trap

**SMIx00000034 (FCH::SMI::capturedvalid)**

Read-only.

\_aliasHOST; SMIx00000034; SMI=FED8\_0200h

Bits	Description
7:4	Reserved.
3:0	<b>capturedvalid</b> . Read-only. Reset: XXXXb. This is the byte valid buffer to signal which byte is captured for the last transaction that caused the SMI. Bit 0 for byte 0, 1 for byte 1, 2 for byte 2, 3 for byte 3

**SMIx00000038 (FCH::SMI::epbif\_aer\_straps)**

Read-write. Reset: 07FE\_FFEh.

\_aliasHOST; SMIx00000038; SMI=FED8\_0200h

Bits	Description
31:28	Reserved.
27	<b>strap_bif_internal_err_en_sb</b> . Read-write. Reset: 0. Internal error enable.
26	<b>strap_bif_poisoned_advisory_nonfatal_a_sb</b> . Read-write. Reset: 1. Poisoned TLP as advisory nonfatal.
25	<b>strap_bif_acs_direct_translated_p2p_sb</b> . Read-write. Reset: 1. ACS direct translated P2P enable.
24	<b>strap_bif_acs_upstream_forwarding_sb</b> . Read-write. Reset: 1. ACS upstream forwarding enable.
23	<b>strap_bif_acs_p2p_completion_redirect_sb</b> . Read-write. Reset: 1. ACS P2P completion redirect enable.
22	<b>strap_bif_acs_p2p_request_redirect_sb</b> . Read-write. Reset: 1. ACS P2P request redirect enable.
21	<b>strap_bif_acs_translation_blocking_sb</b> . Read-write. Reset: 1. ACS translation blocking enable.
20	<b>strap_bif_acs_source_validation_sb</b> . Read-write. Reset: 1. ACS source validation enable.
19	<b>strap_bif_acs_en_sb</b> . Read-write. Reset: 1. ACS enable.
18	<b>strap_bif_first_rcvd_err_log_sb</b> . Read-write. Reset: 1. First received error log.
17	<b>strap_bif_ecrc_check_en_sb</b> . Read-write. Reset: 1. ECRC check enable.
16	<b>strap_bif_ecrc_gen_en_sb</b> . Read-write. Reset: 0. ECRC generate enable.
15	<b>strap_bif_cpl_abort_err_en_sb</b> . Read-write. Reset: 1. Completer abort error enable.
14	<b>strap_bif_rx_ignore_vend0_ur_sb</b> . Read-write. Reset: 1. Ignore Vendor 0 error.
13	<b>strap_bif_rx_ignore_tc_err_sb</b> . Read-write. Reset: 1. Ignore traffic class error.
12	<b>strap_bif_rx_ignore_msg_err_sb</b> . Read-write. Reset: 1. Ignore message error.
11	<b>strap_bif_rx_ignore_max_payload_err_sb</b> . Read-write. Reset: 1. Ignore maximum payload error.
10	<b>strap_bif_rx_ignore_len_mismatch_err_sb</b> . Read-write. Reset: 1. Ignore length mismatch error.
9	<b>strap_bif_rx_ignore_io_ur_err_sb</b> . Read-write. Reset: 1. Ignore IO UR error.
8	<b>strap_bif_rx_ignore_io_err_sb</b> . Read-write. Reset: 1. Ignore IO error.
7	<b>strap_bif_rx_ignore_ep_err_sb</b> . Read-write. Reset: 1. Ignore poisoned TLP error.
6	<b>strap_bif_rx_ignore_cpl_err_sb</b> . Read-write. Reset: 1. Ignore completion error.
5	<b>strap_bif_rx_ignore_cfg_ur_sb</b> . Read-write. Reset: 1. Ignore config. UR error.
4	<b>strap_bif_rx_ignore_cfg_err_sb</b> . Read-write. Reset: 1. Ignore configuration error.
3	<b>strap_bif_rx_ignore_be_err_sb</b> . Read-write. Reset: 1. Ignore byte enable error.
2	<b>strap_bif_err_reporting_dis_sb</b> . Read-write. Reset: 1. Error reporting disable.
1	<b>strap_bif_aer_en_sb</b> . Read-write. Reset: 1. AER enable.
0	<b>strap_bif_sticky_override_s5</b> . Read-write. Reset: 0. When set to 1, values in this register would override straps loaded from EEPROM.

**SMIx0000003C (FCH::SMI::dataerrorstatus)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx0000003C; SMI=FED8\_0200h

Bits	Description
31:8	Reserved.
7	<b>sirq_iochk.</b> Read-write. Reset: 0. Serial Iochk error write 1 to clear it to 0
6	<b>sataperr.</b> Read-write. Reset: 0. SATA controller internal parity error status write 1 to clear it to 0.
5	<b>umi_uncorrectable_err.</b> Read-write. Reset: 0. UMI uncorrectable error status write 1 to clear it to 0.
4	<b>umi_correctable_err.</b> Read-write. Reset: 0. UMI correctable error status write 1 to clear it to 0.
3	<b>abumigppperr.</b> Read-write. Reset: 0. AB/UMI/GPP parity error status write 1 to clear it to 0.
2	<b>internalgppserr.</b> Read-write. Reset: 0. Internal error status: FCH has detected an internal error from upstream bridge write 1 to clear it to 0.
1	<b>internalperr.</b> Read-write. Reset: 0. Internal devices Perr error status write 1 to clear it to 0.
0	<b>internalserr.</b> Read-write. Reset: 0. Internal devices serr error status write 1 to clear it to 0.

**SMIx00000040 (FCH::SMI::scimap0)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000040; SMI=FED8\_0200h

Bits	Description
31:29	Reserved.
28:24	<b>scimap_3.</b> Read-write. Reset: 00h. <b>Description:</b> Mapping of AGPIO22 to one of 32 Event_Status. 00000: map event source 3 to the input of Event_Status bit 0 00001: map event source 3 to the input of Event_Status bit 1 11111: map event source 3 to the input of Event_Status bit 31
23:21	Reserved.
20:16	<b>scimap_2.</b> Read-write. Reset: 00h. <b>Description:</b> Mapping of AGPIO3 to one of 32 Event_Status. 00000: map event source 2 to the input of Event_Status bit 0 00001: map event source 2 to the input of Event_Status bit 1 11111: map event source 2 to the input of Event_Status bit 31
15:13	Reserved.
12:8	<b>scimap_1.</b> Read-write. Reset: 00h. <b>Description:</b> Mapping of GENINT2_L to one of 32 Event_Status. 00000: map event source 1 to the input of Event_Status bit 0 00001: map event source 1 to the input of Event_Status bit 1 11111: map event source 1 to the input of Event_Status bit 31
7:5	Reserved.
4:0	<b>scimap_0.</b> Read-write. Reset: 00h. <b>Description:</b> Mapping of GENINT1_L to one of 32 Event_Status. 00000: map event source 0 to the input of Event_Status bit 0 00001: map event source 0 to the input of Event_Status bit 1 11111: map input event0 to the input of Event_Status bit 31

**SMIx00000044 (FCH::SMI::scimap1)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000044; SMI=FED8\_0200h

Bits	Description
31:29	Reserved.
28:24	<b>scimap_7.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of AGPIO5 to one of 32 Event_Status. 00000: map event source 7 to the input of Event_Status bit 0 00001: map event source 7 to the input of Event_Status bit 1 11111: map event source 7 to the input of Event_Status bit 31
23:21	Reserved.
20:16	<b>scimap_6.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of SPKR to one of 32 Event_Status. 00000: map event source 6 to the input of Event_Status bit 0 00001: map event source 6 to the input of Event_Status bit 1 11111: map event source 6 to the input of Event_Status bit 31
15:13	Reserved.
12:8	<b>scimap_5.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of AGPIO11 to one of 32 Event_Status. 00000: map event source 5 to the input of Event_Status bit 0 00001: map event source 5 to the input of Event_Status bit 1 11111: map event source 5 to the input of Event_Status bit 31
7:5	Reserved.
4:0	<b>scimap_4.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of AGPIO4 to one of 32 Event_Status. 00000: map event source 4 to the input of Event_Status bit 0 00001: map event source 4 to the input of Event_Status bit 1 11111: map event source 4 to the input of Event_Status bit 31

**SMIx00000048 (FCH::SMI::scimap2)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000048; SMI=FED8\_0200h

Bits	Description
31:29	Reserved.
28:24	<b>scimap_11.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of AGPIO7 to one of 32 Event_Status. 00000: map event source 11 to the input of Event_Status bit 0 00001: map event source 11 to the input of Event_Status bit 1 11111: map event source 11 to the input of Event_Status bit 31
23:21	Reserved.
20:16	<b>scimap_10.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of AGPIO6 to one of 32 Event_Status. 00000: map event source 10 to the input of Event_Status bit 0 00001: map event source 10 to the input of Event_Status bit 1 11111: map event source 10 to the input of Event_Status bit 31
15:13	Reserved.
12:8	<b>scimap_9.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of AGPIO29 to one of 32 Event_Status. 00000: map event source 9 to the input of Event_Status bit 0 00001: map event source 9 to the input of Event_Status bit 1 11111: map event source 9 to the input of Event_Status bit 31
7:5	Reserved.
4:0	<b>scimap_8.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of WAKE_L to one of 32 Event_Status. 00000: map event source 8 to the input of Event_Status bit 0 00001: map event source 8 to the input of Event_Status bit 1 11111: map event source 8 to the input of Event_Status bit 31

**SMIx0000004C (FCH::SMI::scimap3)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx0000004C; SMI=FED8\_0200h

Bits	Description
31:29	Reserved.
28:24	<b>scimap_15.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of USB_OC3_L to one of 32 Event_Status. 00000: map event source 15 to the input of Event_Status bit 0 00001: map event source 15 to the input of Event_Status bit 1 11111: map event source 15 to the input of Event_Status bit 31
23:21	Reserved.
20:16	<b>scimap_14.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of USB_OC2_L to one of 32 Event_Status. 00000: map event source 14 to the input of Event_Status bit 0 00001: map event source 14 to the input of Event_Status bit 1 11111: map event source 14 to the input of Event_Status bit 31
15:13	Reserved.
12:8	<b>scimap_13.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of USB_OC1_L to one of 32 Event_Status. 00000: map event source 13 to the input of Event_Status bit 0 00001: map event source 13 to the input of Event_Status bit 1 11111: map event source 13 to the input of Event_Status bit 31
7:5	Reserved.
4:0	<b>scimap_12.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of USB_OC0_L to one of 32 Event_Status. 00000: map event source 12 to the input of Event_Status bit 0 00001: map event source 12 to the input of Event_Status bit 1 11111: map event source 12 to the input of Event_Status bit 31



**SMIx00000050 (FCH::SMI::scimap4)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000050; SMI=FED8\_0200h

Bits	Description
31:29	Reserved.
28:24	<b>scimap_19.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of SYS_RESET_L to one of 32 Event_Status. 00000: map event source 19 to the input of Event_Status bit 0 00001: map event source 19 to the input of Event_Status bit 1 11111: map event source 19 to the input of Event_Status bit 31
23:21	Reserved.
20:16	<b>scimap_18.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of FANIN0 to one of 32 Event_Status. 00000: map event source 18 to the input of Event_Status bit 0 00001: map event source 18 to the input of Event_Status bit 1 11111: map event source 18 to the input of Event_Status bit 31
15:13	Reserved.
12:8	<b>scimap_17.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of AGPIO32 to one of 32 Event_Status. 00000: map event source 17 to the input of Event_Status bit 0 00001: map event source 17 to the input of Event_Status bit 1 11111: map event source 17 to the input of Event_Status bit 31
7:5	Reserved.
4:0	<b>scimap_16.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of AGPIO23 to one of 32 Event_Status. 00000: map event source 16 to the input of Event_Status bit 0 00001: map event source 16 to the input of Event_Status bit 1 11111: map event source 16 to the input of Event_Status bit 31

**SMIx00000054 (FCH::SMI::scimap5)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000054; SMI=FED8\_0200h

Bits	Description
31:29	Reserved.
28:24	<b>scimap_23.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of AGPIO8 to one of 32 Event_Status. 00000: map event source 23 to the input of Event_Status bit 0 00001: map event source 23 to the input of Event_Status bit 1 11111: map event source 23 to the input of Event_Status bit 31
23:21	Reserved.
20:16	<b>scimap_22.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of AGPIO9 to one of 32 Event_Status. 00000: map event source 22 to the input of Event_Status bit 0 00001: map event source 22 to the input of Event_Status bit 1 11111: map event source 22 to the input of Event_Status bit 31
15:13	Reserved.
12:8	<b>scimap_21.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of PWR_BTN_L to one of 32 Event_Status. 00000: map event source 21 to the input of Event_Status bit 0 00001: map event source 21 to the input of Event_Status bit 1 11111: map event source 21 to the input of Event_Status bit 31
7:5	Reserved.
4:0	<b>scimap_20.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of AGPIO40 to one of 32 Event_Status. 00000: map event source 20 to the input of Event_Status bit 0 00001: map event source 20 to the input of Event_Status bit 1 11111: map event source 20 to the input of Event_Status bit 31

**SMIx00000058 (FCH::SMI::scimap6)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000058; SMI=FED8\_0200h

Bits	Description
31:29	Reserved.
28:24	<b>scimap_27.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of eSPI_WAKE_PME (~eSPI_WAKE_PME_B, active high) to one of 32 Event_Status. 00000: map event source 27 to the input of Event_Status bit 0 00001: map event source 27 to the input of Event_Status bit 1 11111: map event source 27 to the input of Event_Status bit 31
23:21	Reserved.
20:16	<b>scimap_26.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of eSPI system event (~eSPI_SYS_EVT_B, active high) to one of 32 Event_Status. 00000: map event source 26 to the input of Event_Status bit 0 00001: map event source 26 to the input of Event_Status bit 1 11111: map event source 26 to the input of Event_Status bit 31
15:13	Reserved.
12:8	<b>scimap_25.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of MP2 GPIO[0] to one of 32 Event_Status. 00000: map event source 25 to the input of Event_Status bit 0 00001: map event source 25 to the input of Event_Status bit 1 11111: map event source 25 to the input of Event_Status bit 31
7:5	Reserved.
4:0	<b>scimap_24.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of MP2 wakeup event to one of 32 Event_Status. 00000: map event source 24 to the input of Event_Status bit 0 00001: map event source 24 to the input of Event_Status bit 1 11111: map event source 24 to the input of Event_Status bit 31

**SMIx0000005C (FCH::SMI::scimap7)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx0000005C; SMI=FED8\_0200h

Bits	Description
31:29	Reserved.
28:24	<b>scimap_31.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of Reserved Sci31 to one of 32 Event_Status. 00000: map event source 31 to the input of Event_Status bit 0 00001: map event source 31 to the input of Event_Status bit 1 11111: map event source 31 to the input of Event_Status bit 31
23:21	Reserved.
20:16	<b>scimap_30.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of NB GPP Hot Plug to one of 32 Event_Status. 00000: map event source 30 to the input of Event_Status bit 0 00001: map event source 30 to the input of Event_Status bit 1 11111: map event source 30 to the input of Event_Status bit 31
15:13	Reserved.
12:8	<b>scimap_29.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of NB GPP_PME to one of 32 Event_Status. 00000: map event source 29 to the input of Event_Status bit 0 00001: map event source 29 to the input of Event_Status bit 1 11111: map event source 29 to the input of Event_Status bit 31
7:5	Reserved.
4:0	<b>scimap_28.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of MP2 GPIO[1] to one of 32 Event_Status. 00000: map event source 28 to the input of Event_Status bit 0 00001: map event source 28 to the input of Event_Status bit 1 11111: map event source 28 to the input of Event_Status bit 31

**SMIx00000060 (FCH::SMI::scimap8)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000060; SMI=FED8\_0200h

Bits	Description
31:29	Reserved.
28:24	<b>scimap_35.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of PM FakeSts2 to one of 32 Event_Status. 00000: map event source 35 to the input of Event_Status bit 0 00001: map event source 35 to the input of Event_Status bit 1 11111: map event source 35 to the input of Event_Status bit 31
23:21	Reserved.
20:16	<b>scimap_34.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of PM FakeSts1 to one of 32 Event_Status. 00000: map event source 34 to the input of Event_Status bit 0 00001: map event source 34 to the input of Event_Status bit 1 11111: map event source 34 to the input of Event_Status bit 31
15:13	Reserved.
12:8	<b>scimap_33.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of PM FakeSts0 to one of 32 Event_Status. 00000: map event source 33 to the input of Event_Status bit 0 00001: map event source 33 to the input of Event_Status bit 1 11111: map event source 33 to the input of Event_Status bit 31
7:5	Reserved.
4:0	<b>scimap_32.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of WakeB(BP_WAKE_L) to one of 32 Event_Status. 00000: map event source 33 to the input of Event_Status bit 0 00001: map event source 33 to the input of Event_Status bit 1 11111: map event source 33 to the input of Event_Status bit 31

**SMIx00000064 (FCH::SMI::scimap9)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000064; SMI=FED8\_0200h

Bits	Description
31:29	Reserved.
28:24	<b>scimap_39.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of AZ_Wake to one of 32 Event_Status. 00000: map event source 39 to the input of Event_Status bit 0 00001: map event source 39 to the input of Event_Status bit 1 11111: map event source 39 to the input of Event_Status bit 31
23:21	Reserved.
20:16	<b>scimap_38.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of USB4_1 PME to one of 32 Event_Status. 00000: map event source 38 to the input of Event_Status bit 0 00001: map event source 38 to the input of Event_Status bit 1 11111: map event source 38 to the input of Event_Status bit 31
15:13	Reserved.
12:8	<b>scimap_37.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of USB4_0 PME to one of 32 Event_Status. 00000: map event source 37 to the input of Event_Status bit 0 00001: map event source 37 to the input of Event_Status bit 1 11111: map event source 37 to the input of Event_Status bit 31
7:5	Reserved.
4:0	<b>scimap_36.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of ESPI SCI (~eSPI_SCI_B, active high) to one of 32 Event_Status. 00000: map event source 37 to the input of Event_Status bit 0 00001: map event source 37 to the input of Event_Status bit 1 11111: map event source 37 to the input of Event_Status bit 31

**SMIx00000068 (FCH::SMI::scimap10)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000068; SMI=FED8\_0200h

Bits	Description
31:29	Reserved.
28:24	<b>scimap_43.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of AltHPET timer event to one of 32 Event_Status. 00000: map event source 43 to the input of Event_Status bit 0 00001: map event source 43 to the input of Event_Status bit 1 11111: map event source 43 to the input of Event_Status bit 31
23:21	Reserved.
20:16	<b>scimap_42.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of USBADP_USB3_0/1(they are OR-ed together) to one of 32 Event_Status. 00000: map event source 42 to the input of Event_Status bit 0 00001: map event source 42 to the input of Event_Status bit 1 11111: map event source 42 to the input of Event_Status bit 31
15:13	Reserved.
12:8	<b>scimap_41.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of Gpio Interrupt to one of 32 Event_Status. 00000: map event source 41 to the input of Event_Status bit 0 00001: map event source 41 to the input of Event_Status bit 1 11111: map event source 41 to the input of Event_Status bit 31
7:5	Reserved.
4:0	<b>scimap_40.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of Reserved Sci40 to one of 32 Event_Status. 00000: map event source 40 to the input of Event_Status bit 0 00001: map event source 40 to the input of Event_Status bit 1 11111: map event source 40 to the input of Event_Status bit 31

**SMIx0000006C (FCH::SMI::scimap11)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx0000006C; SMI=FED8\_0200h

Bits	Description
31:29	Reserved.
28:24	<b>scimap_47.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of SMBUS0 Interrupt event to one of 32 Event_Status. 00000: map event source 47 to the input of Event_Status bit 0 00001: map event source 47 to the input of Event_Status bit 1 11111: map event source 47 to the input of Event_Status bit 31
23:21	Reserved.
20:16	<b>scimap_46.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of I2S wake event to one of 32 Event_Status. 00000: map event source 46 to the input of Event_Status bit 0 00001: map event source 46 to the input of Event_Status bit 1 11111: map event source 46 to the input of Event_Status bit 31
15:13	Reserved.
12:8	<b>scimap_45.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of ASF Master and Slave Interrupt event to one of 32 Event_Status. 00000: map event source 45 to the input of Event_Status bit 0 00001: map event source 45 to the input of Event_Status bit 1 11111: map event source 45 to the input of Event_Status bit 31
7:5	Reserved.
4:0	<b>scimap_44.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of internal FAN/THERMAL event to one of 32 Event_Status. 00000: map event source 44 to the input of Event_Status bit 0 00001: map event source 44 to the input of Event_Status bit 1 11111: map event source 44 to the input of Event_Status bit 31



**SMIx00000070 (FCH::SMI::scimap12)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000070; SMI=FED8\_0200h

Bits	Description
31:29	Reserved.
28:24	<b>scimap_51.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of power button event to one of the 32 Event_Status bits. 00000: map event source 50 to the input of Event_Status bit 0 00001: map event source 50 to the input of Event_Status bit 1 11111: map event source 50 to the input of Event_Status bit 31
23:21	Reserved.
20:16	<b>scimap_50.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of LLB# to one of the 32 Event_Status bits. 00000: map event source 50 to the input of Event_Status bit 0 00001: map event source 50 to the input of Event_Status bit 1 11111: map event source 50 to the input of Event_Status bit 31
15:13	Reserved.
12:8	<b>scimap_49.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of internal traffic monitor to one of 32 Event_Status. 00000: map event source 49 to the input of Event_Status bit 0 00001: map event source 49 to the input of Event_Status bit 1 11111: map event source 49 to the input of Event_Status bit 31
7:5	Reserved.
4:0	<b>scimap_48.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of TWARN pin to one of 32 Event_Status. 00000: map event source 48 to the input of Event_Status bit 0 00001: map event source 48 to the input of Event_Status bit 1 11111: map event source 48 to the input of Event_Status bit 31

**SMIx00000074 (FCH::SMI::scimap13)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000074; SMI=FED8\_0200h

Bits	Description
31:29	Reserved.
28:24	<b>scimap_55.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of RAS_event status to one of the 32 Event_Status bits. 00000: map event source 55 to the input of Event_Status bit 0 00001: map event source 55 to the input of Event_Status bit 1 11111: map event source 55 to the input of Event_Status bit 31
23:21	Reserved.
20:16	<b>scimap_54.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of "SCI assertion message from APU" to one of the 32 Event_Status bits. 00000: map event source 54 to the input of Event_Status bit 0 00001: map event source 54 to the input of Event_Status bit 1 11111: map event source 54 to the input of Event_Status bit 31
15:13	Reserved.
12:8	<b>scimap_53.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of "HW assertion message from APU" to one of the 32 Event_Status bits. 00000: map event source 53 to the input of Event_Status bit 0 00001: map event source 53 to the input of Event_Status bit 1 11111: map event source 53 to the input of Event_Status bit 31
7:5	Reserved.
4:0	<b>scimap_52.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of PROCHOT# pin to one of the 32 Event_Status bits. 00000: map event source 52 to the input of Event_Status bit 0 00001: map event source 52 to the input of Event_Status bit 1 11111: map event source 52 to the input of Event_Status bit 31

**SMIx00000078 (FCH::SMI::scimap14)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000078; SMI=FED8\_0200h

Bits	Description
31:29	Reserved.
28:24	<b>scimap_59.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of Debug State Machine 0 Trigger to one of the 32 Event_Status bits. 00000: map event source 59 to the input of Event_Status bit 0 00001: map event source 59 to the input of Event_Status bit 1 11111: map event source 59 to the input of Event_Status bit 31
23:21	Reserved.
20:16	<b>scimap_58.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of AC/DC timer event to one of the 32 Event_Status bits. 00000: map event source 58 to the input of Event_Status bit 0 00001: map event source 58 to the input of Event_Status bit 1 11111: map event source 58 to the input of Event_Status bit 31
15:13	Reserved.
12:8	<b>scimap_57.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of USB3_HC1 PME event to one of the 32 Event_Status bits. 00000: map event source 57 to the input of Event_Status bit 0 00001: map event source 57 to the input of Event_Status bit 1 11111: map event source 57 to the input of Event_Status bit 31
7:5	Reserved.
4:0	<b>scimap_56.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of USB3_HC0 PME event to one of the 32 Event_Status bits. 00000: map event source 56 to the input of Event_Status bit 0 00001: map event source 56 to the input of Event_Status bit 1 11111: map event source 56 to the input of Event_Status bit 31

**SMIx0000007C (FCH::SMI::scimap15)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx0000007C; SMI=FED8\_0200h

Bits	Description
31:29	Reserved.
28:24	<b>scimap_63.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of Current Temperature Status to one of the 32 Event_Status bits. 00000: map event source 63 to the input of Event_Status bit 0 00001: map event source 63 to the input of Event_Status bit 1 11111: map event source 63 to the input of Event_Status bit 31
23:21	Reserved.
20:16	<b>scimap_62.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of USB3 HC4 PME event to one of the 32 Event_Status bits. 00000: map event source 62 to the input of Event_Status bit 0 00001: map event source 62 to the input of Event_Status bit 1 11111: map event source 62 to the input of Event_Status bit 31
15:13	Reserved.
12:8	<b>scimap_61.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of USB3 HC3 PME event to one of the 32 Event_Status bits. 00000: map event source 61 to the input of Event_Status bit 0 00001: map event source 61 to the input of Event_Status bit 1 11111: map event source 61 to the input of Event_Status bit 31
7:5	Reserved.
4:0	<b>scimap_60.</b> Read-write. Reset: 00h.
	<b>Description:</b> Mapping of Debug State Machine 1 Trigger to one of the 32 Event_Status bits. 00000: map event source 60 to the input of Event_Status bit 0 00001: map event source 60 to the input of Event_Status bit 1 11111: map event source 60 to the input of Event_Status bit 31

**SMIx00000080 (FCH::SMI::smistatus0)**

Read,Write-1-to-clear. Reset: 0000\_0000h.

\_aliasHOST; SMIx000000080; SMI=FED8\_0200h

Bits	Description
31	<b>usbadb_usb3_2_pme</b> . Read,Write-1-to-clear. Reset: 0. Status of Reserved Sci31
30	<b>nbgpphp_event30</b> . Read,Write-1-to-clear. Reset: 0. Status of NB Hot Plug event
29	<b>nbgpppme_event29</b> . Read,Write-1-to-clear. Reset: 0. Status of NB GPP PME
28	<b>smi_event28</b> . Read,Write-1-to-clear. Reset: 0. Status of MP2 GPIO[1]
27	<b>espi_wake_pme_event27</b> . Read,Write-1-to-clear. Reset: 0. Status of ESPI WAKE/PME event
26	<b>smi_event26</b> . Read,Write-1-to-clear. Reset: 0. Status of ~eSPI_SYS_EVT_B
25	<b>smi_event25</b> . Read,Write-1-to-clear. Reset: 0. Status of MP2 GPIO[0]
24	<b>smi_event24</b> . Read,Write-1-to-clear. Reset: 0. Status of MP2 wakeup event
23	<b>gevent23_status_event23</b> . Read,Write-1-to-clear. Reset: 0. Status of AGPIO8
22	<b>gevent22_status_event22</b> . Read,Write-1-to-clear. Reset: 0. Status of AGPIO9
21	<b>gevent21_status_event21</b> . Read,Write-1-to-clear. Reset: 0. Status of PWR_BTN_L
20	<b>gevent20_status_event20</b> . Read,Write-1-to-clear. Reset: 0. Status of AGPIO40
19	<b>gevent19_status_event19</b> . Read,Write-1-to-clear. Reset: 0. Status of SYS_RESET_L
18	<b>gevent18_status_event18</b> . Read,Write-1-to-clear. Reset: 0. Status of FANIN0
17	<b>gevent17_status_event17</b> . Read,Write-1-to-clear. Reset: 0. Status of ESPI_RESET_L
16	<b>gevent16_status_event16</b> . Read,Write-1-to-clear. Reset: 0. Status of AGPIO23
15	<b>gevent15_status_event15</b> . Read,Write-1-to-clear. Reset: 0. Status of USB_OC3_L
14	<b>gevent14_status_event14</b> . Read,Write-1-to-clear. Reset: 0. Status of USB_OC2_L
13	<b>gevent13_status_event13</b> . Read,Write-1-to-clear. Reset: 0. Status of USB_OC1_L
12	<b>gevent12_status_event12</b> . Read,Write-1-to-clear. Reset: 0. Status of USB_OC0_L
11	<b>gevent11_status_event11</b> . Read,Write-1-to-clear. Reset: 0. Status of AGPIO7
10	<b>gevent10_status_event10</b> . Read,Write-1-to-clear. Reset: 0. Status of AGPIO6
9	<b>gevent9_status_event9</b> . Read,Write-1-to-clear. Reset: 0. Status of LPC_SMI_L
8	<b>gevent8_status_event8</b> . Read,Write-1-to-clear. Reset: 0. Status of WAKE_L
7	<b>gevent7_status_event7</b> . Read,Write-1-to-clear. Reset: 0. Status of AGPIO5
6	<b>gevent6_status_event6</b> . Read,Write-1-to-clear. Reset: 0. Status of SPKR
5	<b>gevent5_status_event5</b> . Read,Write-1-to-clear. Reset: 0. Status of LPC_PD_L
4	<b>gevent4_status_event4</b> . Read,Write-1-to-clear. Reset: 0. Status of AGPIO4
3	<b>gevent3_status_event3</b> . Read,Write-1-to-clear. Reset: 0. Status of LPC_PME_L
2	<b>gevent2_status_event2</b> . Read,Write-1-to-clear. Reset: 0. Status of AGPIO3
1	<b>gevent1_status_event1</b> . Read,Write-1-to-clear. Reset: 0. Status of GENINT2_L
0	<b>gevent0_status_event0</b> . Read,Write-1-to-clear. Reset: 0. Status of GENINT1_L

**SMIx00000084 (FCH::SMI::smistatus1)**

Read,Write-1-to-clear. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000084; SMI=FED8\_0200h

Bits	Description
31	<b>current_temperature_status.</b> Read,Write-1-to-clear. Reset: 0. Status of Current Temperature Status
30	<b>usb4p_usb3_1_pme.</b> Read,Write-1-to-clear. Reset: 0. Status of USB3 HC3 PME 4
29	<b>usb4p_usb3_0_pme.</b> Read,Write-1-to-clear. Reset: 0. Status of USB3 HC3 PME 3
28	<b>dsm_trigger_1.</b> Read,Write-1-to-clear. Reset: 0. Status of Debug State Machine Trigger 1
27	<b>dsm_trigger_0.</b> Read,Write-1-to-clear. Reset: 0. Status of Debug State Machine Trigger 0
26	<b>acdctimerevent_event58.</b> Read,Write-1-to-clear. Reset: 0. Status of AcDcTimer wake up event (Wake Alarm Device)
25	<b>usb3_hc1_pme.</b> Read,Write-1-to-clear. Reset: 0. Status of XHC1 PME
24	<b>usb3_hc0_pme.</b> Read,Write-1-to-clear. Reset: 0. Status of USB
23	<b>ras_event55.</b> Read,Write-1-to-clear. Reset: 0. Internal devices SERR error status
22	<b>apusciasrtion_event54.</b> Read,Write-1-to-clear. Reset: 0. Status of APU SCI request
21	<b>apuhwasserrtion_event53.</b> Read,Write-1-to-clear. Reset: 0. Status of APU Hw assertion
20	<b>prochot_event52.</b> Read,Write-1-to-clear. Reset: 0. Status of Prochot event
19	<b>pwrbutton_event51.</b> Read,Write-1-to-clear. Reset: 0. Status of PwrButton (rising edge) writing 1 to clear it to 0.
18	<b>illb_event50.</b> Read,Write-1-to-clear. Reset: 0. Status of iLLB# assertion
17	<b>trafficmonitorintr_event49.</b> Read,Write-1-to-clear. Reset: 0. Status of FCH Traffic Monitor Interrupt
16	<b>twarn_event48.</b> Read,Write-1-to-clear. Reset: 0. Status of FCH TWARN
15	<b>smbus0_event47.</b> Read,Write-1-to-clear. Reset: 0. Status of FCH SMBUS0 Master interrupt
14	<b>i2swake_event46.</b> Read,Write-1-to-clear. Reset: 0. Status of I2S wake event
13	<b>asfrintr_event45.</b> Read,Write-1-to-clear. Reset: 0. Status of FCH ASF Master and Slave interrupt
12	<b>fanthermalgevent_event44.</b> Read,Write-1-to-clear. Reset: 0. Status of FCH FanThermal
11	<b>altmmtimersts_event43.</b> Read,Write-1-to-clear. Reset: 0. Status of AltMmTimer Alarm
10	<b>usb4p_usb3_drd.</b> Read,Write-1-to-clear. Reset: 0. Status of USBADP_USB3_DRD_0/1(They are OR-ed together)
9	<b>gpointr_event41.</b> Read,Write-1-to-clear. Reset: 0. Status of GPIO interrupt
8	<b>usb4_2_pme.</b> Read,Write-1-to-clear. Reset: 0. Status of Reserved Sci40
7	<b>az_wake.</b> Read,Write-1-to-clear. Reset: 0. Status of AZ Wake
6	<b>usb4_1_pme.</b> Read,Write-1-to-clear. Reset: 0. Status of USB4_1_PME
5	<b>usb4_0_pme.</b> Read,Write-1-to-clear. Reset: 0. Status of USB4_0_PME
4	<b>smi_event36.</b> Read,Write-1-to-clear. Reset: 0. Status of ~eSPI_SCI_B
3	<b>fakests2_event35.</b> Read,Write-1-to-clear. Reset: 0. Status of Fake2
2	<b>fakests1_event34.</b> Read,Write-1-to-clear. Reset: 0. Status of Fake1
1	<b>fakests0_event33.</b> Read,Write-1-to-clear. Reset: 0. Status of Fake0
0	<b>smi_event32.</b> Read,Write-1-to-clear. Reset: 0. Status of WAKE_L

**SMIx00000088 (FCH::SMI::smistatus2)**

Read,Write-1-to-clear. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000088; SMI=FED8\_0200h

Bits	Description
31:27	Reserved.
26	<b>emulate64_event90.</b> Read,Write-1-to-clear. Reset: 0. Status of Emulation Io Port 60/64h
25:21	Reserved.
20	<b>pciserr_event84.</b> Read,Write-1-to-clear. Reset: 0. Status of Serr assertion on Pci bus
19	<b>prothot_event83.</b> Read,Write-1-to-clear. Reset: 0. Status of ProtHot event
18	<b>vbatlow_event82.</b> Read,Write-1-to-clear. Reset: 0. Status of VBAT low
17	<b>sim_event81.</b> Read,Write-1-to-clear. Reset: 0. Status of Intruder Alert Status
16:15	Reserved.
14	<b>smbus0intr_event78.</b> Read,Write-1-to-clear. Reset: 0. Status of SMBUS0 interrupt request
13	<b>serialirqsmi_event77.</b> Read,Write-1-to-clear. Reset: 0. Status of Smi request from Serial IRQ
12	<b>usbsmi_event76.</b> Read,Write-1-to-clear. Reset: 0. Status of Usb Smi request
11	<b>smicmdport_event75.</b> Read,Write-1-to-clear. Reset: 0. Status of Writing Smi Command Port
10	<b>pwrbtn_event74.</b> Read,Write-1-to-clear. Reset: 0. Status of Power Button being pressed
9	<b>bios_rls_event73.</b> Read,Write-1-to-clear. Reset: 0. Status of BIOS_RLS
8	<b>gbl_rls_event72.</b> Read,Write-1-to-clear. Reset: 0. Status of GBL event
7:3	Reserved.
2	<b>ial2h_acpi_assertion_event66.</b> Read,Write-1-to-clear. Reset: 0. Status of iAL2H_ACPI_Assertion
1	<b>slp_type_event65.</b> Read,Write-1-to-clear. Reset: 0. Status is set when ACPI SLP_TYP register bit 2 is programmed
0	<b>smi_event64.</b> Read,Write-1-to-clear. Reset: 0. Status of KBRst

**SMIx00000090 (FCH::SMI::smistatus4)**

Read,Write-1-to-clear. Reset: 0000\_0000h.

\_aliasHOST; SMIx00000090; SMI=FED8\_0200h

Bits	Description
31:29	Reserved.
28	<b>cfgtrapping0_event156.</b> Read,Write-1-to-clear. Reset: 0. Status of Pci configuration cycle Trapping0 Smi request
27:25	Reserved.
24	<b>memtrapping0_event152.</b> Read,Write-1-to-clear. Reset: 0. Status of memory Trapping0 Smi request
23	<b>iotrapping3_event151.</b> Read,Write-1-to-clear. Reset: 0. Status of Io Trapping3 Smi request
22	<b>iotrapping2_event150.</b> Read,Write-1-to-clear. Reset: 0. Status of Io Trapping2 Smi request
21	<b>iotrapping1_event149.</b> Read,Write-1-to-clear. Reset: 0. Status of Io Trapping1 Smi request
20	<b>iotrapping0_event148.</b> Read,Write-1-to-clear. Reset: 0. Status of Io Trapping0 Smi request
19	Reserved.
18	<b>espi_smi_event146.</b> Read,Write-1-to-clear. Reset: 0. Status of eSPI SMI event
17	Reserved.
16	<b>absmitrap_event144.</b> Read,Write-1-to-clear. Reset: 0. Status of AB Smi trapping request
15	<b>longtimer_event143.</b> Read,Write-1-to-clear. Reset: 0. Status of Long Timer Smi request
14	<b>shorttimer_event142.</b> Read,Write-1-to-clear. Reset: 0. Status of Short Timer Smi request
13	<b>cf9write_event141.</b> Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 1: Cf9 bit 2 has been written to 1. 0: Cf9 bit 2 is not written to 1.
12:6	Reserved.
5	<b>fanin0sts_event133.</b> Read,Write-1-to-clear. Reset: 0. Status of FanIn0 event
4:0	Reserved.

**SMIx00000094 (FCH::SMI::smipointer)**

Read-only.

This register is meant as a faster mechanism to locate the SMI source. BIOS can examine this register to find out the SMI source instead of reading SmiStatus0 through SmiStatus4 individually

\_aliasHOST; SMIx00000094; SMI=FED8\_0200h

Bits	Description
15:6	Reserved.
5	<b>smistatussource4</b> . Read-only. Reset: 0. Indicate whether the SMI source is from SmiStatus4[31:0] if the corresponding SMI enable is selected.
4	<b>smistatussource3</b> . Read-only. Reset: 0. Indicate whether the SMI source is from SmiStatus3[31:0] if the corresponding SMI enable is selected.
3	<b>smistatussource2</b> . Read-only. Reset: 0. Indicates whether the SMI source is from SmiStatus2[31:0] if the corresponding SMI enable is selected.
2	<b>smistatussource1</b> . Read-only. Reset: 0. Indicates whether the SMI source is from SmiStatus1[31:0] if the corresponding SMI enable is selected.
1	<b>smistatussource0</b> . Read-only. Reset: 0. Indicates whether the SMI source is from SmiStatus0[31:0] if the corresponding SMI enable is selected.
0	<b>smiscisource</b> . Read-only. Reset: 0. Indicates whether the SMI source is from SMISCI.

**SMIx00000096 (FCH::SMI::smi\_short\_long\_timer)**

Read-write. Reset: 0000h.

\*Note: This register 96h can be either "SmiShortTimer" or "SmiLongTimer," depending on the select bit "SmiTimerSel" in SMI\_Reg 98[29]. The default setting (SmiTimerSel=0) selects this register as "SmiShortTimer" software needs to set the "SmiTimerSel=1" to select this register as "SmiLongTimer".

\_aliasHOST; SMIx00000096; SMI=FED8\_0200h

Bits	Description
15	<b>timeren</b> . Read-write. Reset: 0. <b>Description:</b> Enable the SMI short Timer or long timer, which is selected by SmiTimerEn (PMIO_98[29]). 0 = Disable 1 = Enable
14:0	<b>smitimercount</b> . Read-write. Reset: 0000h. <b>Description:</b> Actual timer duration = (TimerTime + 1) * 2us (Short Timer) Actual timer duration = (TimerTime + 1) * 1ms (Long Timer)



**SMIx00000098 (FCH::SMI::smitrig0)**

Read-write. Reset: 8FFF\_FFFFh.

\_aliasHOST; SMIx00000098; SMI=FED8\_0200h

Bits	Description
31	<b>smienb.</b> Read-write. Reset: 1. <b>Description:</b> Enable SMI function. 0: Enable 1: Disable
30	Reserved.
29	<b>smitimersel.</b> Read-write. Reset: 0. <b>Description:</b> 0: Selects the SMI_Reg 96h to be SMIShortTimer register. 1: Selects the SMI_Reg 96h to be SMILongTimer register.
28	<b>eos.</b> Read-write. Reset: 0. This bit is set to 1 by SW to enable SMI generation. It is cleared by hardware after a SMI event has occurred. When Eos is clear, subsequent pending SMI event will be blocked.
27	<b>fakests2.</b> Read-write. Reset: 1. Program the value to emulate an SMI input event.
26	<b>fakests1.</b> Read-write. Reset: 1. Program the value to emulate an SMI input event.
25	<b>fakests0.</b> Read-write. Reset: 1. Program the value to emulate an SMI input event.
24	<b>trappingirqonpic.</b> Read-write. Reset: 1. <b>Description:</b> SMI will be generated when 0: Trapping Irq0 ~ 15 of IoAPIC 1: Trapping Irq0 ~ 15 of PIC
23	<b>smitrig23.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
22	<b>smitrig22.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
21	<b>smitrig21.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
20	<b>smitrig20.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
19	<b>smitrig19.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
18	<b>smitrig18.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
17	<b>smitrig17.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
16	<b>smitrig16.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
15	<b>smitrig15.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
14	<b>smitrig14.</b> Read-write. Reset: 1.

	<b>Description:</b> 1: Active high 0: Active low
13	<b>smitrig13.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
12	<b>smitrig12.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
11	<b>smitrig11.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
10	<b>smitrig10.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
9	<b>smitrig9.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
8	<b>smitrig8.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
7	<b>smitrig7.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
6	<b>smitrig6.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
5	<b>smitrig5.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
4	<b>smitrig4.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
3	<b>smitrig3.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
2	<b>smitrig2.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
1	<b>smitrig1.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
0	<b>smitrig0.</b> Read-write. Reset: 1. <b>Description:</b> This defines the trigger mode for SmiStatus0[23:0]. Note these are different from SciTrig 0: Active low 1: Active high

**SMIx0000009C (FCH::SMI::smitrig1)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx0000009C; SMI=FED8\_0200h

Bits	Description
31	<b>smiirq31trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
30	<b>smiirq30trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
29	<b>smiirq29trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
28	<b>smiirq28trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
27	<b>smiirq27trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
26	<b>smiirq26trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
25	<b>smiirq25trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
24	<b>smiirq24trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
23	<b>smiirq23trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
22	<b>smiirq22trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
21	<b>smiirq21trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
20	<b>smiirq20trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
19	<b>smiirq19trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
18	<b>smiirq18trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
17	<b>smiirq17trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
16	<b>smiirq16trig.</b> Read-write. Reset: 0.

	<b>Description:</b> 0: Active low 1: Active high
15	<b>smiirq15trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
14	<b>smiirq14trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
13	<b>smiirq13trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
12	<b>smiirq12trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
11	<b>smiirq11trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
10	<b>smiirq10trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
9	<b>smiirq9trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
8	<b>smiirq8trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
7	<b>smiirq7trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
6	<b>smiirq6trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
5	<b>smiirq5trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
4	<b>smiirq4trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
3	<b>smiirq3trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
2	<b>smiirq2trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
1	<b>smiirq1trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high
0	<b>smiirq0trig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Active low 1: Active high

**SMIx000000A0 (FCH::SMI::smicontrol0)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx000000A0; SMI=FED8\_0200h

Bits	Description
31:30	<b>smicontrol_15.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT15 00: Disable 01: SMI 10: NMI 11: IRQ13
29:28	<b>smicontrol_14.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT14 00: Disable 01: SMI 10: NMI 11: IRQ13
27:26	<b>smicontrol_13.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT13 00: Disable 01: SMI 10: NMI 11: IRQ13
25:24	<b>smicontrol_12.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT12 00: Disable 01: SMI 10: NMI 11: IRQ13
23:22	<b>smicontrol_11.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT11 00: Disable 01: SMI 10: NMI 11: IRQ13
21:20	<b>smicontrol_10.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT10 00: Disable 01: SMI 10: NMI 11: IRQ13
19:18	<b>smicontrol_9.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT9 00: Disable 01: SMI 10: NMI 11: IRQ13
17:16	<b>smicontrol_8.</b> Read-write. Reset: 0h.

	<b>Description:</b> Control for GEVENT8 00: Disable 01: SMI 10: NMI 11: IRQ13
15:14	<b>smicontrol_7.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT7 00: Disable 01: SMI 10: NMI 11: IRQ13
13:12	<b>smicontrol_6.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT6 00: Disable 01: SMI 10: NMI 11: IRQ13
11:10	<b>smicontrol_5.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT5 00: Disable 01: SMI 10: NMI 11: IRQ13
9:8	<b>smicontrol_4.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT4 00: Disable 01: SMI 10: NMI 11: IRQ13
7:6	<b>smicontrol_3.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT3 00: Disable 01: SMI 10: NMI 11: IRQ13
5:4	<b>smicontrol_2.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT2 00: Disable 01: SMI 10: NMI 11: IRQ13
3:2	<b>smicontrol_1.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT1 00: Disable 01: SMI 10: NMI 11: IRQ13
1:0	<b>smicontrol_0.</b> Read-write. Reset: 0h.

	<b>Description:</b> Control for GEVENT0 00: Disable 01: SMI 10: NMI 11: IRQ13
--	---

SMIx000000A4 (FCH::SMI::smicontrol1)	
Read-write. Reset: 0000_0000h.	
_aliasHOST; SMIx000000A4; SMI=FED8_0200h	
Bits	Description
31:30	<b>smicontrol_31.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GPP_PME (devie 21, function3) 00: Disable 01: SMI 10: NMI 11: IRQ13
29:28	<b>smicontrol_30.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GPP_PME (devie 21, function2) 00: Disable 01: SMI 10: NMI 11: IRQ13
27:26	<b>smicontrol_29.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GPP_PME (devie 21, function1) 00: Disable 01: SMI 10: NMI 11: IRQ13
25:24	<b>smicontrol_28.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GPP_PME (devie 21, function0~3) 00: Disable 01: SMI 10: NMI 11: IRQ13
23:18	Reserved.
17:16	<b>smicontrol_24.</b> Read-write. Reset: 0h. <b>Description:</b> Control for USB_PME (devie 18) 00: Disable 01: SMI 10: NMI 11: IRQ13
15:14	<b>smicontrol_23.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT23 00: Disable 01: SMI 10: NMI 11: IRQ13
13:12	<b>smicontrol_22.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT22 00: Disable 01: SMI 10: NMI 11: IRQ13
11:10	<b>smicontrol_21.</b> Read-write. Reset: 0h.



	<b>Description:</b> Control for GEVENT21 00: Disable 01: SMI 10: NMI 11: IRQ13
9:8	<b>smicontrol_20.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT20 00: Disable 01: SMI 10: NMI 11: IRQ13
7:6	<b>smicontrol_19.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT19 00: Disable 01: SMI 10: NMI 11: IRQ13
5:4	<b>smicontrol_18.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT18 00: Disable 01: SMI 10: NMI 11: IRQ13
3:2	<b>smicontrol_17.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT17 00: Disable 01: SMI 10: NMI 11: IRQ13
1:0	<b>smicontrol_16.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GEVENT16 00: Disable 01: SMI 10: NMI 11: IRQ13

**SMIx000000A8 (FCH::SMI::smicontrol2)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx000000A8; SMI=FED8\_0200h

Bits	Description
31:30	<b>smicontrol_47.</b> Read-write. Reset: 0h. <b>Description:</b> Control for SMBUS0 interrupt 00: Disable 01: SMI 10: NMI 11: IRQ13
29:28	<b>smicontrol_46.</b> Read-write. Reset: 0h. <b>Description:</b> Control for ASF Slave interrupt 00: Disable 01: SMI 10: NMI 11: IRQ13
27:26	<b>smicontrol_45.</b> Read-write. Reset: 0h. <b>Description:</b> Control for ASF Master interrupt 00: Disable 01: SMI 10: NMI 11: IRQ13
25:24	<b>smicontrol_44.</b> Read-write. Reset: 0h. <b>Description:</b> Control for FanThermal Gevent 00: Disable 01: SMI 10: NMI 11: IRQ13
23:22	<b>smicontrol_43.</b> Read-write. Reset: 0h. <b>Description:</b> Control for ALTHPET_TimerSts 00: Disable 01: SMI 10: NMI 11: IRQ13
21:20	<b>smicontrol_42.</b> Read-write. Reset: 0h. <b>Description:</b> Control for CIR PME 00: Disable 01: SMI 10: NMI 11: IRQ13
19:18	<b>smicontrol_41.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GPIO interrupt 00: Disable 01: SMI 10: NMI 11: IRQ13
17:16	<b>smicontrol_40.</b> Read-write. Reset: 0h.

	<b>Description:</b> Control for Ec Gevent0 00: Disable 01: SMI 10: NMI 11: IRQ13
15:14	Reserved.
13:12	<b>smicontrol_38.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Sata Gevent1 00: Disable 01: SMI 10: NMI 11: IRQ13
11:10	<b>smicontrol_37.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Sata Gevent0 00: Disable 01: SMI 10: NMI 11: IRQ13
9:8	<b>smicontrol_36.</b> Read-write. Reset: 0h. <b>Description:</b> Control for PME frpm OTG 00: Disable 01: SMI 10: NMI 11: IRQ13
7:6	<b>smicontrol_35.</b> Read-write. Reset: 0h. <b>Description:</b> Control for FakeSts2 00: Disable 01: SMI 10: NMI 11: IRQ13
5:4	<b>smicontrol_34.</b> Read-write. Reset: 0h. <b>Description:</b> Control for FakeSts1 00: Disable 01: SMI 10: NMI 11: IRQ13
3:2	<b>smicontrol_33.</b> Read-write. Reset: 0h. <b>Description:</b> Control for FakeSts0 00: Disable 01: SMI 10: NMI 11: IRQ13
1:0	<b>smicontrol_32.</b> Read-write. Reset: 0h. <b>Description:</b> Control for GPP_HotPlug (device 21, function 0~3) 00: Disable 01: SMI 10: NMI 11: IRQ13

SMIx000000AC (FCH::SMI::smicontrol3)	
Read-write. Reset: 0000_0000h.	
_aliasHOST; SMIx000000AC; SMI=FED8_0200h	
Bits	Description
31:30	<b>smicontrol_63.</b> Read-write. Reset: 0h. <b>Description:</b> Control for TempTsi event 00: Disable 01: SMI 10: NMI 11: IRQ13
29:28	<b>smicontrol_62.</b> Read-write. Reset: 0h. <b>Description:</b> Control for DSM Cross Trigger event 3 00: Disable 01: SMI 10: NMI 11: IRQ13
27:26	<b>smicontrol_61.</b> Read-write. Reset: 0h. <b>Description:</b> Control for DSM Cross Trigger event 2 00: Disable 01: SMI 10: NMI 11: IRQ13
25:24	<b>smicontrol_60.</b> Read-write. Reset: 0h. <b>Description:</b> Control for DSM Cross Trigger event 1 00: Disable 01: SMI 10: NMI 11: IRQ13
23:22	<b>smicontrol_59.</b> Read-write. Reset: 0h. <b>Description:</b> Control for DSM Cross Trigger event 0 00: Disable 01: SMI 10: NMI 11: IRQ13
21:20	<b>smicontrol_58.</b> Read-write. Reset: 0h. <b>Description:</b> Control for AcDcTimer wake up event (Wake Device in ACPI4.0) 00: Disable 01: SMI 10: NMI 11: IRQ13
19:18	<b>smicontrol_57.</b> Read-write. Reset: 0h. <b>Description:</b> Control for XHC1 (dev 16, func 1) PME 00: Disable 01: SMI 10: NMI 11: IRQ13
17:16	<b>smicontrol_56.</b> Read-write. Reset: 0h.

	<b>Description:</b> Control for XHC0 (dev 16, func 0) PME 00: Disable 01: SMI 10: NMI 11: IRQ13
15:14	<b>smicontrol_55.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Internal devices SERR error status 00: Disable 01: SMI 10: NMI 11: IRQ13
13:12	<b>smicontrol_54.</b> Read-write. Reset: 0h. <b>Description:</b> Control for APU SCI request 00: Disable 01: SMI 10: NMI 11: IRQ13
11:10	<b>smicontrol_53.</b> Read-write. Reset: 0h. <b>Description:</b> Control for APU Hw assertion 00: Disable 01: SMI 10: NMI 11: IRQ13
9:8	<b>smicontrol_52.</b> Read-write. Reset: 0h. <b>Description:</b> Control for ProcHot event 00: Disable 01: SMI 10: NMI 11: IRQ13
7:6	<b>smicontrol_51.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Power button event 00: Disable 01: SMI 10: NMI 11: IRQ13
5:4	<b>smicontrol_50.</b> Read-write. Reset: 0h. <b>Description:</b> Control for iLLB# 00: Disable 01: SMI 10: NMI 11: IRQ13
3:2	<b>smicontrol_49.</b> Read-write. Reset: 0h. <b>Description:</b> Control for internal Traffic monitor interrupt 00: Disable 01: SMI 10: NMI 11: IRQ13
1:0	<b>smicontrol_48.</b> Read-write. Reset: 0h.

	<b>Description:</b> Control for TWARN# 00: Disable 01: SMI 10: NMI 11: IRQ13
--	--

**SMIx000000B0 (FCH::SMI::smicontrol4)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx000000B0; SMI=FED8\_0200h

Bits	Description
31:30	<b>smicontrol_79.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Ec Smi request0 00: Disable 01: SMI 10: NMI 11: IRQ13
29:28	<b>smicontrol_78.</b> Read-write. Reset: 0h. <b>Description:</b> Control for SMBUS0 interrupt 00: Disable 01: SMI 10: NMI 11: IRQ13
27:26	<b>smicontrol_77.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Smi request form serial Irq 00: Disable 01: SMI 10: NMI 11: IRQ13
25:24	<b>smicontrol_76.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Usb Smi request 00: Disable 01: SMI 10: NMI 11: IRQ13
23:22	<b>smicontrol_75.</b> Read-write. Reset: 0h. <b>Description:</b> Control for writing Smi command port 00: Disable 01: SMI 10: NMI 11: IRQ13
21:20	<b>smicontrol_74.</b> Read-write. Reset: 0h. <b>Description:</b> Control for power button being pressed 00: Disable 01: SMI 10: NMI 11: IRQ13
19:18	<b>smicontrol_73.</b> Read-write. Reset: 0h. <b>Description:</b> Control for writing BIOS_RLS 00: Disable 01: SMI 10: NMI 11: IRQ13
17:16	<b>smicontrol_72.</b> Read-write. Reset: 0h.

	<b>Description:</b> Control for writing GBL_RLS 00: Disable 01: SMI 10: NMI 11: IRQ13
15:8	Reserved.
7:6	<b>smicontrol_67.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Sata AHCI event 00: Disable 01: SMI 10: NMI 11: IRQ13
5:4	<b>smicontrol_66.</b> Read-write. Reset: 0h. <b>Description:</b> Control for iAL2H_ACPI_Assertion 00: Disable 01: SMI 10: NMI 11: IRQ13
3:2	<b>smicontrol_65.</b> Read-write. Reset: 0h. <b>Description:</b> Control for writing SLP_TYP to put the system in S state. 00: Disable 01: SMI 10: NMI 11: IRQ13
1:0	<b>smicontrol_64.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Keyboard Reset event. 00: Disable 01: SMI 10: NMI 11: IRQ13



**SMIx000000B4 (FCH::SMI::smicontrol5)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx000000B4; SMI=FED8\_0200h

Bits	Description
31:28	Reserved.
27:26	<b>smicontrol_93.</b> Read-write. Reset: 0h. <b>Description:</b> Control for HD audio FLR 00: Disable 01: SMI 10: NMI 11: IRQ13
25:24	<b>smicontrol_92.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Sata FLR 00: Disable 01: SMI 10: NMI 11: IRQ13
23:22	<b>smicontrol_91.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Usb FLR 00: Disable 01: SMI 10: NMI 11: IRQ13
21:20	<b>smicontrol_90.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Emulation64 00: Disable 01: SMI 10: NMI 11: IRQ13
19:18	<b>smicontrol_89.</b> Read-write. Reset: 0h. <b>Description:</b> Control for ThermalTrip# assertion 00: Disable 01: SMI 10: NMI 11: IRQ13
17:12	Reserved.
11:10	<b>smicontrol_85.</b> Read-write. Reset: 0h. <b>Description:</b> Control for SB GPP Serr#(device 21, function 0~3) 01: SMI 10: NMI 11: IRQ13
9:8	<b>smicontrol_84.</b> Read-write. Reset: 0h. <b>Description:</b> Control for SERR# 00: Disable 01: SMI 10: NMI 11: IRQ13
7:6	<b>smicontrol_83.</b> Read-write. Reset: 0h.

	<b>Description:</b> Control for ProcHot 00: Disable 01: SMI 10: NMI 11: IRQ13
5:4	<b>smicontrol_82.</b> Read-write. Reset: 0h. <b>Description:</b> Control for VBAT low 00: Disable 01: SMI 10: NMI 11: IRQ13
3:2	<b>smicontrol_81.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Intruder event. 00: Disable 01: SMI 10: NMI 11: IRQ13
1:0	<b>smicontrol_80.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Ec Smi request1 00: Disable 01: SMI 10: NMI 11: IRQ13

**SMIx000000B8 (FCH::SMI::smicontrol6)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx000000B8; SMI=FED8\_0200h

Bits	Description
31:0	Reserved.

**SMIx000000BC (FCH::SMI::smicontrol7)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx000000BC; SMI=FED8\_0200h

Bits	Description
31:0	Reserved.

SMIx000000C0 (FCH::SMI::smicontrol8)	
Read-write. Reset: 0000_0000h.	
_aliasHOST; SMIx000000C0; SMI=FED8_0200h	
Bits	Description
31:30	<b>smicontrol_143.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Long timer 00: Disable 01: SMI 10: NMI 11: IRQ13
29:28	<b>smicontrol_142.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Short timer 00: Disable 01: SMI 10: NMI 11: IRQ13
27:26	<b>smicontrol_141.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Cf9 Io write 00: Disable 01: SMI 10: NMI 11: IRQ13
25:24	<b>smicontrol_140.</b> Read-write. Reset: 0h. <b>Description:</b> Control for FakeSts2 00: Disable 01: SMI 10: NMI 11: IRQ13 Note: FakeSts2 defined in PMIO can be programmed to generate SMI/NMI/IRQ13 specified in those two bits.
23:22	<b>smicontrol_139.</b> Read-write. Reset: 0h. <b>Description:</b> Control for FakeSts1 00: Disable 01: SMI 10: NMI 11: IRQ13 Note: FakeSts1 defined in PMIO can be programmed to generate SMI/NMI/IRQ13 specified in those two bits.
21:20	<b>smicontrol_138.</b> Read-write. Reset: 0h. <b>Description:</b> Control for FakeSts0 00: Disable 01: SMI 10: NMI 11: IRQ13 Note: FakeSts0 defined in PMIO can be programmed to generate SMI/NMI/IRQ13 specified in those two bits.
19:12	Reserved.
11:10	<b>smicontrol_133.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Fan Tach 0 too slow event 00: Disable 01: SMI 10: NMI 11: IRQ13
9:0	Reserved.

**SMIx000000C4 (FCH::SMI::smicontrol9)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx000000C4; SMI=FED8\_0200h

Bits	Description
31:26	Reserved.
25:24	<b>smicontrol_156.</b> Read-write. Reset: 0h. <b>Description:</b> Control for configuration cycle trapping 0 00: Disable 01: SMI 10: NMI 11: IRQ13
23:18	Reserved.
17:16	<b>smicontrol_152.</b> Read-write. Reset: 0h. <b>Description:</b> Control for memory trapping 0 00: Disable 01: SMI 10: NMI 11: IRQ13
15:14	<b>smicontrol_151.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Io trapping 3 00: Disable 01: SMI 10: NMI 11: IRQ13
13:12	<b>smicontrol_150.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Io trapping 2 00: Disable 01: SMI 10: NMI 11: IRQ13
11:10	<b>smicontrol_149.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Io trapping 1 00: Disable 01: SMI 10: NMI 11: IRQ13
9:8	<b>smicontrol_148.</b> Read-write. Reset: 0h. <b>Description:</b> Control for Io trapping 0 00: Disable 01: SMI 10: NMI 11: IRQ13
7:4	Reserved.
3:2	<b>smicontrol_145.</b> Read-write. Reset: 0h. <b>Description:</b> Control for P state message 0 00: Disable 01: SMI 10: NMI 11: IRQ13
1:0	<b>smicontrol_144.</b> Read-write. Reset: 0h.

	<b>Description:</b> Control for AB Smi trapping request 00: Disable 01: SMI 10: NMI 11: IRQ13
--	---

#### SMIx000000C8 (FCH::SMI::smilevel0)

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx000000C8; SMI=FED8\_0200h

Bits	Description
31	<b>smilevel31.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
30	<b>smilevel30.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
29	<b>smilevel29.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
28	<b>smilevel28.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
27	<b>smilevel27.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
26	<b>smilevel26.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
25	<b>smilevel25.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
24	<b>smilevel24.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
23	<b>smilevel23.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
22	<b>smilevel22.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
21	<b>smilevel21.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
20	<b>smilevel20.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
19	<b>smilevel19.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
18	<b>smilevel18.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
17	<b>smilevel17.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
16	<b>smilevel16.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
15	<b>smilevel15.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
14	<b>smilevel14.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
13	<b>smilevel13.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
12	<b>smilevel12.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
11	<b>smilevel11.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
10	<b>smilevel10.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
9	<b>smilevel9.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
8	<b>smilevel8.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
7	<b>smilevel7.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
6	<b>smilevel6.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
5	<b>smilevel5.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
4	<b>smilevel4.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
3	<b>smilevel3.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
2	<b>smilevel2.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
1	<b>smilevel1.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
0	<b>smilevel0.</b> Read-write. Reset: 0. <b>Description:</b> This defines the edge/level trigger mode for SmiLevel[31:0]. Note these are different from SciLevl 0: Edge trigger mode, 1: Level trigger mode

**SMIx000000CC (FCH::SMI::smileveltrig0)**

Read-write. Reset: FFFF\_FFFFh.

\_aliasHOST; SMIx000000CC; SMI=FED8\_0200h

Bits	Description
31	<b>smitrig31.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
30	<b>smitrig30.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
29	<b>smitrig29.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
28	<b>smitrig28.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
27	<b>smitrig27.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
26	<b>smitrig26.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
25	<b>smitrig25.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
24	<b>smitrig24.</b> Read-write. Reset: 1. <b>Description:</b> This defines the trigger mode for SmiTrig[31:24]. 1: Active high 0: Active low
23:0	Reserved.

**SMIx000000D0 (FCH::SMI::smilevel1)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx000000D0; SMI=FED8\_0200h

Bits	Description
31	<b>smilevel63.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
30	<b>smilevel62.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
29	<b>smilevel61.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
28	<b>smilevel60.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
27	<b>smilevel59.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
26	<b>smilevel58.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
25	<b>smilevel57.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
24	<b>smilevel56.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
23	<b>smilevel55.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
22	<b>smilevel54.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
21	<b>smilevel53.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
20	<b>smilevel52.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
19	<b>smilevel51.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
18	<b>smilevel50.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
17	<b>smilevel49.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
16	<b>smilevel48.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
15	<b>smilevel47.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
14	<b>smilevel46.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
13	<b>smilevel45.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
12	<b>smilevel44.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
11	<b>smilevel43.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
10	<b>smilevel42.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
9	<b>smilevel41.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
8	<b>smilevel40.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
7	<b>smilevel39.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
6	<b>smilevel38.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
5	<b>smilevel37.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
4	<b>smilevel36.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
3	<b>smilevel35.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
2	<b>smilevel34.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
1	<b>smilevel33.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
0	<b>smilevel32.</b> Read-write. Reset: 0. <b>Description:</b> This defines the edge/level trigger mode for SmiLevel[63:32]. Note these are different from SciLevl 0: Edge trigger mode, 1: Level trigger mode

**SMIx000000D4 (FCH::SMI::smilevel2)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx000000D4; SMI=FED8\_0200h

Bits	Description
31	<b>smilevel95.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
30	<b>smilevel94.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
29	<b>smilevel93.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
28	<b>smilevel92.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
27	<b>smilevel91.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
26	<b>smilevel90.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
25	<b>smilevel89.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
24	<b>smilevel88.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
23	<b>smilevel87.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
22	<b>smilevel86.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
21	<b>smilevel85.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
20	<b>smilevel84.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
19	<b>smilevel83.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
18	<b>smilevel82.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
17	<b>smilevel81.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
16	<b>smilevel80.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
15	<b>smilevel79.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
14	<b>smilevel78.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
13	<b>smilevel77.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
12	<b>smilevel76.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
11	<b>smilevel75.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
10	<b>smilevel74.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
9	<b>smilevel73.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
8	<b>smilevel72.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
7	<b>smilevel71.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
6	<b>smilevel70.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
5	<b>smilevel69.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
4	<b>smilevel68.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
3	<b>smilevel67.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
2	<b>smilevel66.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
1	<b>smilevel65.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
0	<b>smilevel64.</b> Read-write. Reset: 0. <b>Description:</b> This defines the edge/level trigger mode for SmiLevel[95:64]. Note these are different from SciLevl 0: Edge trigger mode, 1: Level trigger mode



**SMIx000000D8 (FCH::SMI::smilevel3)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx000000D8; SMI=FED8\_0200h

Bits	Description
31	<b>smilevel127.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
30	<b>smilevel126.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
29	<b>smilevel125.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
28	<b>smilevel124.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
27	<b>smilevel123.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
26	<b>smilevel122.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
25	<b>smilevel121.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
24	<b>smilevel120.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
23	<b>smilevel119.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
22	<b>smilevel118.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
21	<b>smilevel117.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
20	<b>smilevel116.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
19	<b>smilevel115.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
18	<b>smilevel114.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
17	<b>smilevel113.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
16	<b>smilevel112.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
15	<b>smilevel111.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
14	<b>smilevel110.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
13	<b>smilevel109.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
12	<b>smilevel108.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
11	<b>smilevel107.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
10	<b>smilevel106.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
9	<b>smilevel105.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
8	<b>smilevel104.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
7	<b>smilevel103.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
6	<b>smilevel102.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
5	<b>smilevel101.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
4	<b>smilevel100.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
3	<b>smilevel99.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
2	<b>smilevel98.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
1	<b>smilevel97.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
0	<b>smilevel96.</b> Read-write. Reset: 0. <b>Description:</b> This defines the edge/level trigger mode for SmiLevel[127:96]. Note these are different from SciLevel 0: Edge trigger mode, 1: Level trigger mode

**SMIx000000DC (FCH::SMI::smilevel4)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; SMIx000000DC; SMI=FED8\_0200h

Bits	Description
31	<b>smilevel159.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
30	<b>smilevel158.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
29	<b>smilevel157.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
28	<b>smilevel156.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
27	<b>smilevel155.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
26	<b>smilevel154.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
25	<b>smilevel153.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
24	<b>smilevel152.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
23	<b>smilevel151.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
22	<b>smilevel150.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
21	<b>smilevel149.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
20	<b>smilevel148.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
19	<b>smilevel147.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
18	<b>smilevel146.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
17	<b>smilevel145.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
16	<b>smilevel144.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
15	<b>smilevel143.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
14	<b>smilevel142.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
13	<b>smilevel141.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
12	<b>smilevel140.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
11	<b>smilevel139.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
10	<b>smilevel138.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
9	<b>smilevel137.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
8	<b>smilevel136.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
7	<b>smilevel135.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
6	<b>smilevel134.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
5	<b>smilevel133.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
4	<b>smilevel132.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
3	<b>smilevel131.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
2	<b>smilevel130.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
1	<b>smilevel129.</b> Read-write. Reset: 0. 0: Edge trigger mode, 1: Level trigger mode
0	<b>smilevel128.</b> Read-write. Reset: 0. <b>Description:</b> This defines the edge/level trigger mode for SmiLevel[159:128]. Note these are different from SciLevel 0: Edge trigger mode, 1: Level trigger mode

**SMIx000000E0 (FCH::SMI::smitrig2)**

Read-write. Reset: FFFF\_FFFFh.

\_aliasHOST; SMIx000000E0; SMI=FED8\_0200h

Bits	Description
31	<b>smitrig63.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
30	<b>smitrig62.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
29	<b>smitrig61.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
28	<b>smitrig60.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
27	<b>smitrig59.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
26	<b>smitrig58.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
25	<b>smitrig57.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
24	<b>smitrig56.</b> Read-write. Reset: 1. <b>Description:</b> 1: Active high 0: Active low
23	<b>smitrig55.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
22	<b>smitrig54.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
21	<b>smitrig53.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
20	<b>smitrig52.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
19	<b>smitrig51.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
18	<b>smitrig50.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
17	<b>smitrig49.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
16	<b>smitrig48.</b> Read-write. Reset: 1.

	<b>Description:</b> 0: Active low 1: Active high
15	<b>smitrig47.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
14	<b>smitrig46.</b> Read-write. Reset: 1. <b>Description:</b> 1: Active high 0: Active low
13	<b>smitrig45.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
12	<b>smitrig44.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
11	<b>smitrig43.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
10	<b>smitrig42.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
9	<b>smitrig41.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
8	<b>smitrig40.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
7	<b>smitrig39.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
6	<b>smitrig38.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
5	<b>smitrig37.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
4	<b>smitrig36.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
3	<b>smitrig35.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
2	<b>smitrig34.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
1	<b>smitrig33.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
0	<b>smitrig32.</b> Read-write. Reset: 1.

	<b>Description:</b> This defines the trigger mode for SmiTrig[63:32]. 0: Active low 1: Active high
--	--

**SMIx000000E4 (FCH::SMI::smitrig3)**

Read-write. Reset: FFFF\_FFFFh.

\_aliasHOST; SMIx000000E4; SMI=FED8\_0200h

Bits	Description
31	<b>smitrig95.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
30	<b>smitrig94.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
29	<b>smitrig93.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
28	<b>smitrig92.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
27	<b>smitrig91.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
26	<b>smitrig90.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
25	<b>smitrig89.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
24	<b>smitrig88.</b> Read-write. Reset: 1. <b>Description:</b> 1: Active high 0: Active low
23	<b>smitrig87.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
22	<b>smitrig86.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
21	<b>smitrig85.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
20	<b>smitrig84.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
19	<b>smitrig83.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
18	<b>smitrig82.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
17	<b>smitrig81.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
16	<b>smitrig80.</b> Read-write. Reset: 1.

	<b>Description:</b> 0: Active low 1: Active high
15	<b>smitrig79.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
14	<b>smitrig78.</b> Read-write. Reset: 1. <b>Description:</b> 1: Active high 0: Active low
13	<b>smitrig77.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
12	<b>smitrig76.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
11	<b>smitrig75.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
10	<b>smitrig74.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
9	<b>smitrig73.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
8	<b>smitrig72.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
7	<b>smitrig71.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
6	<b>smitrig70.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
5	<b>smitrig69.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
4	<b>smitrig68.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
3	<b>smitrig67.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
2	<b>smitrig66.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
1	<b>smitrig65.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
0	<b>smitrig64.</b> Read-write. Reset: 1.

	<div><div><b>Description:</b> This defines the trigger mode for SmiTrig[95:64].</div><div>0: Active low</div><div>1: Active high</div></div>
--	--



**SMIx000000E8 (FCH::SMI::smitrig4)**

Read-write. Reset: FFFF\_FFFFh.

\_aliasHOST; SMIx000000E8; SMI=FED8\_0200h

Bits	Description
31	<b>smitrig159.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
30	<b>smitrig158.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
29	<b>smitrig157.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
28	<b>smitrig156.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
27	<b>smitrig155.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
26	<b>smitrig154.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
25	<b>smitrig153.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
24	<b>smitrig152.</b> Read-write. Reset: 1. <b>Description:</b> 1: Active high 0: Active low
23	<b>smitrig151.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
22	<b>smitrig150.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
21	<b>smitrig149.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
20	<b>smitrig148.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
19	<b>smitrig147.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
18	<b>smitrig146.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
17	<b>smitrig145.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
16	<b>smitrig144.</b> Read-write. Reset: 1.

	<b>Description:</b> 0: Active low 1: Active high
15	<b>smitrig143.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
14	<b>smitrig142.</b> Read-write. Reset: 1. <b>Description:</b> 1: Active high 0: Active low
13	<b>smitrig141.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
12	<b>smitrig140.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
11	<b>smitrig139.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
10	<b>smitrig138.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
9	<b>smitrig137.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
8	<b>smitrig136.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
7	<b>smitrig135.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
6	<b>smitrig134.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
5	<b>smitrig133.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
4	<b>smitrig132.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
3	<b>smitrig131.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
2	<b>smitrig130.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
1	<b>smitrig129.</b> Read-write. Reset: 1. <b>Description:</b> 0: Active low 1: Active high
0	<b>smitrig128.</b> Read-write. Reset: 1.

	<b>Description:</b> This defines the trigger mode for SmiTrig[159:128]. 0: Active low 1: Active high
--	--

### 7.3.4 High Precision Event Timer (HPET) Registers

HPET registers are accessed through two methods:

- Memory access to HPET memory address range from FED0\_0000h to FED0\_01FFh. Program PMx00[HpetEn]=1 to enable HPET decoding.
- Memory mapped access through the AcpiMmio region. The HPET registers range from FED8\_0000h+C00h to FED8\_0000h+CFFh. See PMx04[MmioEn].

All registers in this block are reset by PciRstB, which will be asserted in the following conditions:

- Resume Reset: This reset is asserted in G3 state and deasserted during G3 to S5 transition.
- System Reset: From system reset button.
- S0 Reset events: some events that happen in S0 state, such as CF9 and Keyboard Reset.
- Sleep states: S3, S5 and Power saving mode states.

HPETx00000000 (FCH::TMR::HPET::id)	
Read-only. Reset: 1022_8201h.	
_aliasHOST; HPETx00000000; HPET=FED0_0000h	
Bits	Description
31:16	<b>vendorid.</b> Read-only. Reset: 1022h. AMD vendor ID.
15	<b>legacy_cap.</b> Read-only. Reset: 1. Legacy replacement interrupt is supported.
14	Reserved.
13	<b>counter_size_cap.</b> Read-only. Reset: 0. <b>Description:</b> Main counter is 32-bits wide or 64-bit mode. The read only register value depend on the HPET_width_sel register bit: When HPET_width_sel is 0, Counter_Size_Cap=0, means 32-bits wide When HPET_width_sel is 1, Counter_Size_Cap=1, means 64-bits wide
12:8	<b>num_tmr_cap.</b> Read-only. Reset: 02h. Three timers are supported.
7:0	<b>revid.</b> Read-only. Reset: 01h. Revision ID.

HPETx00000004 (FCH::TMR::HPET::clkperiod)	
Read-only. Reset: 0429_B17Eh.	
_aliasHOST; HPETx00000004; HPET=FED0_0000h	
Bits	Description
31:0	<b>counter_clk_period.</b> Read-only. Reset: 0429_B17Eh. Specifies the clock period of each HPET timer tick. HPET main counter runs at 14.31818/48 MHz. The unit is femptoseconds ( $10^{-15}$ seconds). Note: The value of this register can be modified through MISC_Reg: 34h.

**HPETx00000010 (FCH::TMR::HPET::hpetconfig)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; HPETx00000010; HPET=FED0\_0000h

Bits	Description
31:2	Reserved.
1	<b>legacyen.</b> Read-write. Reset: 0. <b>Description:</b> If LegacyEn is set to 1b then: Timer0 interrupt goes to IRQ0 of PIC controller, INT2 of IoAPIC Timer1 interrupt goes to IRQ8 of PIC controller, INT8 of IoAPIC.
0	<b>tmren.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pause main counter and disable all timer interrupts. 1: Allow main counter to run and allow timer interrupts if enabled.

**HPETx00000020 (FCH::TMR::HPET::interrupt\_status)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; HPETx00000020; HPET=FED0\_0000h

Bits	Description
31:3	Reserved.
2	<b>tmr2intrsts.</b> Read-write. Reset: 0. <b>Description:</b> 0: Timer2 interrupt is not active. 1: Timer2 interrupt is active. Write 1 to clear if timer2 is set to level-triggered mode. When set to edge-triggered mode, software should ignore this bit and always write 0b to this bit.
1	<b>tmr1intrsts.</b> Read-write. Reset: 0. <b>Description:</b> 0: Timer1 interrupt is not active. 1: Timer1 interrupt is active. Write 1 to clear if timer1 is set to level-triggered mode. When set to edge-triggered mode, software should ignore this bit and always write 0b to this bit.
0	<b>tmr0intrsts.</b> Read-write. Reset: 0. <b>Description:</b> 0: Timer0 interrupt is not active. 1: Timer0 interrupt is active. Write 1 to clear if timer0 is set to level-triggered mode. When set to edge-triggered mode, software should ignore this bit and always write 0b to this bit.

**HPETx000000F0 (FCH::TMR::HPET::main\_counter\_l)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; HPETx000000F0; HPET=FED0\_0000h

Bits	Description
31:0	<b>maincounter_l.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> HPET main counter, increment by 1 on every clock. Counter should be written to only when halted. The width of main counter depends on HPET_width_sel register bit: 0: HPET main counter is 32-bit. MainCounter_L is the valid counter bits. MainCounter_H is not valid. 1: HPET main counter is 64-bit. Both MainCounter_L and MainCounter_H are valid.

**HPETx000000F4 (FCH::TMR::HPET::main\_counter\_h)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; HPETx000000F4; HPET=FED0\_0000h

Bits	Description
31:0	<b>maincounter_h.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> HPET main counter, increment by 1 on every clock. Counter should be written to only when halted. The width of main counter depends on HPET_width_sel register bit: 0: HPET main counter is 32-bit. MainCounter_L is the valid counter bits. MainCounter_H is not valid. 1: HPET main counter is 64-bit. Both MainCounter_L and MainCounter_H are valid.

**HPETx00000100 (FCH::TMR::HPET::tmr0\_conf\_cap\_l)**

Read-write. Reset: 0000\_8010h.

Hardware supports 3 timers, &lt;N&gt; is 0, 1 or 2.

\_aliasHOST; HPETx00000100; HPET=FED0\_0000h

Bits	Description
31:16	Reserved.
15	<b>tmrfsbcap.</b> Read-write. Reset: 1. FSB delivery is supported. Read only.
14	<b>tmrfsben.</b> Read-write. Reset: 0. Set to 1 to enable FSB (Front Side Bus) delivery of interrupt.
13:9	<b>tmrintroute.</b> Read-write. Reset: 00h. These 5 bits specify which INT entry of IoAPIC the timer is routed to when LegacyEn is not set.
8	<b>tmr32modeen.</b> Read-write. Reset: 0. Timer n 32-bit Mode: (where n is the timer number: 0 to 2). Software can set this read/write bit to force a 64-bit timer to behave as a 32-bit timer. This is typically needed if the software is not willing to halt the main counter to read or write a particular timer, and the software is not capable of doing an atomic 64-bit read to the timer. If the timer is not 64 bits wide, then this bit will always be read as 0 and writes will have no effect.
7	Reserved.
6	<b>tmrsetper.</b> Read-write. Reset: 0. <b>Description:</b> Set to 1 to allow software to set the timer's accumulator if the timer is set to periodic mode. The bit is automatically cleared when 'Comparator' is written by software.
5	<b>tmrsizecap.</b> Read-write. Reset: 0. <b>Description:</b> The timer is 32-bits wide/or 64bit wide. This read only register value depend on the HPET_width_sel register bit: When HPET_width_sel is 0, TmrSizeCap =0, means 32-bits wide When HPET_width_sel is 1, TmrSizeCap =1, means 64-bits wide
4	<b>tmrtypcap.</b> Read-write. Reset: 1. The timer supports periodic interrupt delivery mode. Read only.
3	<b>tmrtyp.</b> Read-write. Reset: 0. <b>Description:</b> Select the timer interrupt type: 0: Non-periodic 1: Periodic
2	<b>tmrinten.</b> Read-write. Reset: 0. Set to 1 to enable timer interrupt.
1	<b>tmrinttyp.</b> Read-write. Reset: 0. <b>Description:</b> Control timer interrupt polarity: 0: Edge triggered 1: Level triggered
0	Reserved.

**HPETx00000104 (FCH::TMR::HPET::tmr0\_conf\_cap\_h)**

Read-only. Reset: 00C0\_0000h.

Hardware supports 3 timers, &lt;N&gt; is 0, 1 or 2.

\_aliasHOST; HPETx00000104; HPET=FED0\_0000h

Bits	Description
31:0	<b>tmrintroutecap.</b> Read-only. Reset: 00C0_0000h. Indicates which INT entry of IoAPIC can be assigned to the timer interrupt. Read only.

**HPETx00000108 (FCH::TMR::HPET::tmr0\_comp\_l)**

Read-write. Reset: FFFF\_FFFFh.

Hardware supports 3 timers, &lt;N&gt; is 0, 1 or 2.

\_aliasHOST; HPETx00000108; HPET=FED0\_0000h

Bits	Description
31:0	<b>comparator_l.</b> Read-write. Reset: FFFF_FFFFh. <b>Description:</b> The timer comparator. In non-periodic mode: 'Comparator' is writeable. In periodic mode: 'Comparator' can be modified after TmrSetPer is set to 1. 'Comparator' is periodically incremented by the value last written to this register. By default, value is incremented by 0xFFFFFFFF for 32-bit mode or 0xFFFFFFFFFFFFFFFF for 64-bit mode. The width of Comparator depends on HPET_width_sel register bit: 0: The Comparator is 32-bit. Comparator_L is the valid bits. Comparator_H is not valid. 1: The Comparator is 64-bit. Both Comparator_L and Comparator_H are valid.

**HPETx0000010C (FCH::TMR::HPET::tmr0\_comp\_h)**

Read-write. Reset: 0000\_0000h.

Hardware supports 3 timers, &lt;N&gt; is 0, 1 or 2.

Note1: The HPET spec requires different default values for 32-bit and 64-bit HPET. For 32-bit HPET, Comparator\_H default should be 00000000h. For 64-bit HPET, Comparator\_H default should be FFFFFFFFh. Our HPET can be configured as 32-bit or 64-bit by HPET\_width\_sel register bit. By default, HPET is 32-bit, so the default of Comparator\_H is 00000000h.

\_aliasHOST; HPETx0000010C; HPET=FED0\_0000h

Bits	Description
31:0	<b>comparator_h.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> The timer comparator. In non-periodic mode: 'Comparator' is writeable. In periodic mode: 'Comparator' can be modified after TmrSetPer is set to 1. 'Comparator' is periodically incremented by the value last written to this register. By default, value is incremented by 0xFFFFFFFF for 32-bit mode or 0xFFFFFFFFFFFFFFFF for 64-bit mode. The width of Comparator depends on HPET_width_sel register bit: 0: The Comparator is 32-bit. Comparator_L is the valid bits. Comparator_H is not valid. 1: The Comparator is 64-bit. Both Comparator_L and Comparator_H are valid.

**HPETx00000110 (FCH::TMR::HPET::tmr0\_fsbintval)**

Read-write. Reset: 0000\_0000h.

Hardware supports 3 timers, &lt;N&gt; is 0, 1 or 2.

\_aliasHOST; HPETx00000110; HPET=FED0\_0000h

Bits	Description
31:0	<b>tn_fsb_int_val.</b> Read-write. Reset: 0000_0000h. Software sets this 32-bit field to specify the write data of FSB Interrupt Message.

**HPETx00000114 (FCH::TMR::HPET::tmr0\_fsbintaddr)**

Read-write. Reset: 0000\_0000h.

Hardware supports 3 timers, &lt;N&gt; is 0, 1 or 2.

\_aliasHOST; HPETx00000114; HPET=FED0\_0000h

Bits	Description
31:0	<b>tn_fsb_int_addr.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> Software sets this 32-bit field to specify the address of FSB interrupt Message.

**HPETx00000120 (FCH::TMR::HPET::tmr1\_conf\_cap\_l)**

Read-write. Reset: 0000\_8010h.

Hardware supports 3 timers, &lt;N&gt; is 0, 1 or 2.

\_aliasHOST; HPETx00000120; HPET=FED0\_0000h

Bits	Description
31:16	Reserved.
15	<b>tmrfsbcap.</b> Read-write. Reset: 1. FSB delivery is supported. Read only.
14	<b>tmrfsben.</b> Read-write. Reset: 0. Set to 1 to enable FSB (Front Side Bus) delivery of interrupt.
13:9	<b>tmrintroute.</b> Read-write. Reset: 00h. These 5 bits specify which INT entry of IoAPIC the timer is routed to when LegacyEn is not set.
8	<b>tmr32modeen.</b> Read-write. Reset: 0. Timer n 32-bit Mode: (where n is the timer number: 0 to 2). Software can set this read/write bit to force a 64-bit timer to behave as a 32-bit timer. This is typically needed if the software is not willing to halt the main counter to read or write a particular timer, and the software is not capable of doing an atomic 64-bit read to the timer. If the timer is not 64 bits wide, then this bit will always be read as 0 and writes will have no effect.
7	Reserved.
6	<b>tmrsetper.</b> Read-write. Reset: 0. <b>Description:</b> Set to 1 to allow software to set the timer's accumulator if the timer is set to periodic mode. The bit is automatically cleared when 'Comparator' is written by software.
5	<b>tmrsizecap.</b> Read-write. Reset: 0. <b>Description:</b> The timer is 32-bits wide/or 64bit wide. This read only register value depend on the HPET_width_sel register bit: When HPET_width_sel is 0, TmrSizeCap =0, means 32-bits wide When HPET_width_sel is 1, TmrSizeCap =1, means 64-bits wide
4	<b>tmrtypcap.</b> Read-write. Reset: 1. The timer supports periodic interrupt delivery mode. Read only.
3	<b>tmrtyp.</b> Read-write. Reset: 0. <b>Description:</b> Select the timer interrupt type: 0: Non-periodic 1: Periodic
2	<b>tmrinten.</b> Read-write. Reset: 0. Set to 1 to enable timer interrupt.
1	<b>tmrinttyp.</b> Read-write. Reset: 0. <b>Description:</b> Control timer interrupt polarity: 0: Edge triggered 1: Level triggered
0	Reserved.

**HPETx00000124 (FCH::TMR::HPET::tmr1\_conf\_cap\_h)**

Read-only. Reset: 00C0\_0000h.

Hardware supports 3 timers, &lt;N&gt; is 0, 1 or 2.

\_aliasHOST; HPETx00000124; HPET=FED0\_0000h

Bits	Description
31:0	<b>tmrintroutecap.</b> Read-only. Reset: 00C0_0000h. Indicates which INT entry of IoAPIC can be assigned to the timer interrupt. Read only.

**HPETx00000128 (FCH::TMR::HPET::tmr1\_comp\_l)**

Read-write. Reset: FFFF\_FFFFh.

Hardware supports 3 timers, &lt;N&gt; is 0, 1 or 2.

\_aliasHOST; HPETx00000128; HPET=FED0\_0000h

Bits	Description
31:0	<b>comparator_l.</b> Read-write. Reset: FFFF_FFFFh. <b>Description:</b> The timer comparator. In non-periodic mode: 'Comparator' is writeable. In periodic mode: 'Comparator' can be modified after TmrSetPer is set to 1. 'Comparator' is periodically incremented by the value last written to this register. By default, value is incremented by 0xFFFFFFFF for 32-bit mode or 0xFFFFFFFFFFFFFFFF for 64-bit mode. The width of Comparator depends on HPET_width_sel register bit: 0: The Comparator is 32-bit. Comparator_L is the valid bits. Comparator_H is not valid. 1: The Comparator is 64-bit. Both Comparator_L and Comparator_H are valid.

**HPETx0000012C (FCH::TMR::HPET::tmr1\_comp\_h)**

Read-write. Reset: 0000\_0000h.

Hardware supports 3 timers, &lt;N&gt; is 0, 1 or 2.

Note1: The HPET spec requires different default values for 32-bit and 64-bit HPET. For 32-bit HPET, Comparator\_H default should be 00000000h. For 64-bit HPET, Comparator\_H default should be FFFFFFFFh. Our HPET can be configured as 32-bit or 64-bit by HPET\_width\_sel register bit. By default, HPET is 32-bit, so the default of Comparator\_H is 00000000h.

\_aliasHOST; HPETx0000012C; HPET=FED0\_0000h

Bits	Description
31:0	<b>comparator_h.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> The timer comparator. In non-periodic mode: 'Comparator' is writeable. In periodic mode: 'Comparator' can be modified after TmrSetPer is set to 1. 'Comparator' is periodically incremented by the value last written to this register. By default, value is incremented by 0xFFFFFFFF for 32-bit mode or 0xFFFFFFFFFFFFFFFF for 64-bit mode. The width of Comparator depends on HPET_width_sel register bit: 0: The Comparator is 32-bit. Comparator_L is the valid bits. Comparator_H is not valid. 1: The Comparator is 64-bit. Both Comparator_L and Comparator_H are valid.

**HPETx00000130 (FCH::TMR::HPET::tmr1\_fsbintval)**

Read-write. Reset: 0000\_0000h.

Hardware supports 3 timers, &lt;N&gt; is 0, 1 or 2.

\_aliasHOST; HPETx00000130; HPET=FED0\_0000h

Bits	Description
31:0	<b>tn_fsb_int_val.</b> Read-write. Reset: 0000_0000h. Software sets this 32-bit field to specify the write data of FSB Interrupt Message.



**HPETx00000134 (FCH::TMR::HPET::tmr1\_fsbintaddr)**

Read-write. Reset: 0000\_0000h.

Hardware supports 3 timers, &lt;N&gt; is 0, 1 or 2.

\_aliasHOST; HPETx00000134; HPET=FED0\_0000h

Bits	Description
31:0	<b>tn_fsb_int_addr.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> Software sets this 32-bit field to specify the address of FSB interrupt Message.

**HPETx00000140 (FCH::TMR::HPET::tmr2\_conf\_cap\_l)**

Read-write. Reset: 0000\_8010h.

Hardware supports 3 timers, &lt;N&gt; is 0, 1 or 2.

\_aliasHOST; HPETx00000140; HPET=FED0\_0000h

Bits	Description
31:16	Reserved.
15	<b>tmrfsbcap.</b> Read-write. Reset: 1. FSB delivery is supported. Read only.
14	<b>tmrfsben.</b> Read-write. Reset: 0. Set to 1 to enable FSB (Front Side Bus) delivery of interrupt.
13:9	<b>tmrintroute.</b> Read-write. Reset: 00h. These 5 bits specify which INT entry of IoAPIC the timer is routed to when LegacyEn is not set.
8	<b>tmr32modeen.</b> Read-write. Reset: 0. Timer n 32-bit Mode: (where n is the timer number: 0 to 2). Software can set this read/write bit to force a 64-bit timer to behave as a 32-bit timer. This is typically needed if the software is not willing to halt the main counter to read or write a particular timer, and the software is not capable of doing an atomic 64-bit read to the timer. If the timer is not 64 bits wide, then this bit will always be read as 0 and writes will have no effect.
7	Reserved.
6	<b>tmrsetper.</b> Read-write. Reset: 0. <b>Description:</b> Set to 1 to allow software to set the timer's accumulator if the timer is set to periodic mode. The bit is automatically cleared when 'Comparator' is written by software.
5	<b>tmrsizecap.</b> Read-write. Reset: 0. <b>Description:</b> The timer is 32-bits wide/or 64bit wide. This read only register value depend on the HPET_width_sel register bit: When HPET_width_sel is 0, TmrSizeCap =0, means 32-bits wide When HPET_width_sel is 1, TmrSizeCap =1, means 64-bits wide
4	<b>tmrtypcap.</b> Read-write. Reset: 1. The timer supports periodic interrupt delivery mode. Read only.
3	<b>tmrtyp.</b> Read-write. Reset: 0. <b>Description:</b> Select the timer interrupt type: 0: Non-periodic 1: Periodic
2	<b>tmrinten.</b> Read-write. Reset: 0. Set to 1 to enable timer interrupt.
1	<b>tmrinttyp.</b> Read-write. Reset: 0. <b>Description:</b> Control timer interrupt polarity: 0: Edge triggered 1: Level triggered
0	Reserved.

**HPETx00000144 (FCH::TMR::HPET::tmr2\_conf\_cap\_h)**

Read-only. Reset: 00C0\_0000h.

Hardware supports 3 timers, &lt;N&gt; is 0, 1 or 2.

\_aliasHOST; HPETx00000144; HPET=FED0\_0000h

Bits	Description
31:0	<b>tmrintroutecap.</b> Read-only. Reset: 00C0_0000h. Indicates which INT entry of IoAPIC can be assigned to the timer interrupt. Read only.

**HPETx00000148 (FCH::TMR::HPET::tmr2\_comp\_l)**

Read-write. Reset: FFFF\_FFFFh.

Hardware supports 3 timers, &lt;N&gt; is 0, 1 or 2.

\_aliasHOST; HPETx00000148; HPET=FED0\_0000h

Bits	Description
31:0	<b>comparator_l.</b> Read-write. Reset: FFFF_FFFFh. <b>Description:</b> The timer comparator. In non-periodic mode: 'Comparator' is writeable. In periodic mode: 'Comparator' can be modified after TmrSetPer is set to 1. 'Comparator' is periodically incremented by the value last written to this register. By default, value is incremented by 0xFFFFFFFF for 32-bit mode or 0xFFFFFFFFFFFFFFFF for 64-bit mode. The width of Comparator depends on HPET_width_sel register bit: 0: The Comparator is 32-bit. Comparator_L is the valid bits. Comparator_H is not valid. 1: The Comparator is 64-bit. Both Comparator_L and Comparator_H are valid.

**HPETx0000014C (FCH::TMR::HPET::tmr2\_comp\_h)**

Read-write. Reset: 0000\_0000h.

Hardware supports 3 timers, &lt;N&gt; is 0, 1 or 2.

Note1: The HPET spec requires different default values for 32-bit and 64-bit HPET. For 32-bit HPET, Comparator\_H default should be 00000000h. For 64-bit HPET, Comparator\_H default should be FFFFFFFFh. Our HPET can be configured as 32-bit or 64-bit by HPET\_width\_sel register bit. By default, HPET is 32-bit, so the default of Comparator\_H is 00000000h.

\_aliasHOST; HPETx0000014C; HPET=FED0\_0000h

Bits	Description
31:0	<b>comparator_h.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> The timer comparator. In non-periodic mode: 'Comparator' is writeable. In periodic mode: 'Comparator' can be modified after TmrSetPer is set to 1. 'Comparator' is periodically incremented by the value last written to this register. By default, value is incremented by 0xFFFFFFFF for 32-bit mode or 0xFFFFFFFFFFFFFFFF for 64-bit mode. The width of Comparator depends on HPET_width_sel register bit: 0: The Comparator is 32-bit. Comparator_L is the valid bits. Comparator_H is not valid. 1: The Comparator is 64-bit. Both Comparator_L and Comparator_H are valid.

**HPETx00000150 (FCH::TMR::HPET::tmr2\_fsbintval)**

Read-write. Reset: 0000\_0000h.

Hardware supports 3 timers, &lt;N&gt; is 0, 1 or 2.

\_aliasHOST; HPETx00000150; HPET=FED0\_0000h

Bits	Description
31:0	<b>tn_fsb_int_val.</b> Read-write. Reset: 0000_0000h. Software sets this 32-bit field to specify the write data of FSB Interrupt Message.

**HPETx00000154 (FCH::TMR::HPET::tmr2\_fsbintaddr)**

Read-write. Reset: 0000\_0000h.

Hardware supports 3 timers, &lt;N&gt; is 0, 1 or 2.

\_aliasHOST; HPETx00000154; HPET=FED0\_0000h

Bits	Description
31:0	<b>tn_fsb_int_addr.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> Software sets this 32-bit field to specify the address of FSB interrupt Message.

**HPETx000001B0 (FCH::TMR::HPET::tmr0\_comp\_base\_shadow\_l)**

Read-write. Reset: FFFF\_FFFFh.

\_aliasHOST; HPETx000001B0; HPET=FED0\_0000h

Bits	Description
31:0	<b>tmr0_comp_base_shadow_l.</b> Read-write. Reset: FFFF_FFFFh. <b>Description:</b> This is shadow of the base value of Tmr0 Comparator (HPET_Reg:108h). Reading this register returns the base value of Timer 0 Comparator. Writing the register will change the base value of Timer 0 Comparator.

**HPETx000001B4 (FCH::TMR::HPET::tmr0\_comp\_base\_shadow\_h)**

Read-write. Reset: 0000\_0000h.

Note1: The HPET spec requires different default values for 32-bit and 64-bit HPET. For 32-bit HPET, Comparator\_H default should be 00000000h. For 64-bit HPET, Comparator\_H default should be FFFFFFFFh. Our HPET can be configured as 32-bit or 64-bit by HPET\_width\_sel register bit. By default, HPET is 32-bit, so the default of Comparator\_H is 00000000h.

\_aliasHOST; HPETx000001B4; HPET=FED0\_0000h

Bits	Description
31:0	<b>tmr0_comp_base_shadow_h.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> This is shadow of the base value of Tmr0 Comparator (HPET_Reg:108h). Reading this register returns the base value of Timer 0 Comparator. Writing the register will change the base value of Timer 0 Comparator.

**HPETx000001B8 (FCH::TMR::HPET::tmr0\_comp\_shadow\_l)**

Read-write. Reset: FFFF\_FFFFh.

\_aliasHOST; HPETx000001B8; HPET=FED0\_0000h

Bits	Description
31:0	<b>tmr0_comp_shadow_l.</b> Read-write. Reset: FFFF_FFFFh. <b>Description:</b> This is shadow of the current value of Tmr0 Comparator (HPET_Reg:108h). Reading this register returns the base value of Timer 0 Comparator. Writing the register will change the base value of Timer 0 Comparator.

**HPETx000001BC (FCH::TMR::HPET::tmr0\_comp\_shadow\_h)**

Read-write. Reset: 0000\_0000h.

Note1: The HPET spec requires different default values for 32-bit and 64-bit HPET. For 32-bit HPET, Comparator\_H default should be 00000000h. For 64-bit HPET, Comparator\_H default should be FFFFFFFFh. Our HPET can be configured as 32-bit or 64-bit by HPET\_width\_sel register bit. By default, HPET is 32-bit, so the default of Comparator\_H is 00000000h.

\_aliasHOST; HPETx000001BC; HPET=FED0\_0000h

Bits	Description
31:0	<b>tmr0_comp_shadow_h.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> This is shadow of the current value of Tmr0 Comparator (HPET_Reg:108h). Reading this register returns the base value of Timer 0 Comparator. Writing the register will change the base value of Timer 0 Comparator.

**HPETx000001C0 (FCH::TMR::HPET::tmr1\_comp\_base\_shadow\_l)**

Read-write. Reset: FFFF\_FFFFh.

\_aliasHOST; HPETx000001C0; HPET=FED0\_0000h

Bits	Description
31:0	<b>tmr1_comp_base_shadow_l.</b> Read-write. Reset: FFFF_FFFFh. <b>Description:</b> This is shadow of the base value of Tmr1 Comparator (HPET_Reg:108h). Reading this register returns the base value of Timer 1 Comparator. Writing the register will change the base value of Timer 1 Comparator.

**HPETx000001C4 (FCH::TMR::HPET::tmr1\_comp\_base\_shadow\_h)**

Read-write. Reset: 0000\_0000h.

Note1: The HPET spec requires different default values for 32-bit and 64-bit HPET. For 32-bit HPET, Comparator\_H default should be 00000000h. For 64-bit HPET, Comparator\_H default should be FFFFFFFFh. Our HPET can be configured as 32-bit or 64-bit by HPET\_width\_sel register bit. By default, HPET is 32-bit, so the default of Comparator\_H is 00000000h.

\_aliasHOST; HPETx000001C4; HPET=FED0\_0000h

Bits	Description
31:0	<b>tmr1_comp_base_shadow_h.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> This is shadow of the base value of Tmr1 Comparator (HPET_Reg:108h). Reading this register returns the base value of Timer 1 Comparator. Writing the register will change the base value of Timer 1 Comparator.

**HPETx000001C8 (FCH::TMR::HPET::tmr1\_comp\_shadow\_l)**

Read-write. Reset: FFFF\_FFFFh.

\_aliasHOST; HPETx000001C8; HPET=FED0\_0000h

Bits	Description
31:0	<b>tmr1_comp_shadow_l.</b> Read-write. Reset: FFFF_FFFFh. <b>Description:</b> This is shadow of the current value of Tmr1 Comparator (HPET_Reg:108h). Reading this register returns the base value of Timer 1 Comparator. Writing the register will change the base value of Timer 1 Comparator.

**HPETx000001CC (FCH::TMR::HPET::tmr1\_comp\_shadow\_h)**

Read-write. Reset: 0000\_0000h.

Note1: The HPET spec requires different default values for 32-bit and 64-bit HPET. For 32-bit HPET, Comparator\_H default should be 00000000h. For 64-bit HPET, Comparator\_H default should be FFFFFFFFh. Our HPET can be configured as 32-bit or 64-bit by HPET\_width\_sel register bit. By default, HPET is 32-bit, so the default of Comparator\_H is 00000000h.

\_aliasHOST; HPETx000001CC; HPET=FED0\_0000h

Bits	Description
31:0	<b>tmr1_comp_shadow_h.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> This is shadow of the current value of Tmr1 Comparator (HPET_Reg:108h). Reading this register returns the base value of Timer 1 Comparator. Writing the register will change the base value of Timer 1 Comparator.

**HPETx000001D0 (FCH::TMR::HPET::tmr2\_comp\_base\_shadow\_l)**

Read-write. Reset: FFFF\_FFFFh.

\_aliasHOST; HPETx000001D0; HPET=FED0\_0000h

Bits	Description
31:0	<b>tmr2_comp_base_shadow_l.</b> Read-write. Reset: FFFF_FFFFh. <b>Description:</b> This is shadow of the base value of Tmr2 Comparator (HPET_Reg:108h). Reading this register returns the base value of Timer 2 Comparator. Writing the register will change the base value of Timer 2 Comparator.

**HPETx000001D4 (FCH::TMR::HPET::tmr2\_comp\_base\_shadow\_h)**

Read-write. Reset: 0000\_0000h.

Note1: The HPET spec requires different default values for 32-bit and 64-bit HPET. For 32-bit HPET, Comparator\_H default should be 00000000h. For 64-bit HPET, Comparator\_H default should be FFFFFFFFh. Our HPET can be configured as 32-bit or 64-bit by HPET\_width\_sel register bit. By default, HPET is 32-bit, so the default of Comparator\_H is 00000000h.

\_aliasHOST; HPETx000001D4; HPET=FED0\_0000h

Bits	Description
31:0	<b>tmr2_comp_base_shadow_h.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> This is shadow of the base value of Tmr2 Comparator (HPET_Reg:108h). Reading this register returns the base value of Timer 2 Comparator. Writing the register will change the base value of Timer 2 Comparator.

**HPETx000001D8 (FCH::TMR::HPET::tmr2\_comp\_shadow\_l)**

Read-write. Reset: FFFF\_FFFFh.

\_aliasHOST; HPETx000001D8; HPET=FED0\_0000h

Bits	Description
31:0	<b>tmr2_comp_shadow_l.</b> Read-write. Reset: FFFF_FFFFh. <b>Description:</b> This is shadow of the current value of Tmr2 Comparator (HPET_Reg:108h). Reading this register returns the base value of Timer 2 Comparator. Writing the register will change the base value of Timer 2 Comparator.

**HPETx000001DC (FCH::TMR::HPET::tmr2\_comp\_shadow\_h)**

Read-write. Reset: 0000\_0000h.

Note1: The HPET spec requires different default values for 32-bit and 64-bit HPET. For 32-bit HPET, Comparator\_H default should be 00000000h. For 64-bit HPET, Comparator\_H default should be FFFFFFFFh. Our HPET can be configured as 32-bit or 64-bit by HPET\_width\_sel register bit. By default, HPET is 32-bit, so the default of Comparator\_H is 00000000h.

\_aliasHOST; HPETx000001DC; HPET=FED0\_0000h

Bits	Description
31:0	<b>tmr2_comp_shadow_h.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> This is shadow of the current value of Tmr2 Comparator (HPET_Reg:108h). Reading this register returns the base value of Timer 2 Comparator. Writing the register will change the base value of Timer 2 Comparator.

**HPETx000001E0 (FCH::TMR::HPET::main\_counter\_rtc\_l)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; HPETx000001E0; HPET=FED0\_0000h

Bits	Description
31:0	<b>main_counter_rtc_l.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> This is a shadow of Main_Counter register. It samples the value of Main_Counter at every falling edge of Rtc 32KHz clock for SW to read. When SW writes this register, Main_Counter (offset F0) will be updated with the same value written to this register, and then enabled counting at the next RtcClk falling edge. The purpose of this register is for the convenience of SW save/restore HPET. For 32-bit HPET mode, only Main_Counter_rtc_L is valid. For 64-bit HPET mode, both Main_Counter_rtc_L and Main_Counter_rtc_H are valid.

**HPETx000001E4 (FCH::TMR::HPET::main\_counter\_rtc\_h)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; HPETx000001E4; HPET=FED0\_0000h

Bits	Description
31:0	<b>main_counter_rtc_h.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> This is a shadow of Main_Counter register. It samples the value of Main_Counter at every falling edge of Rtc 32KHz clock for SW to read. When SW writes this register, Main_Counter (offset F0) will be updated with the same value written to this register, and then enabled counting at the next RtcClk falling edge. The purpose of this register is for the convenience of SW save/restore HPET. For 32-bit HPET mode, only Main_Counter_rtc_L is valid. For 64-bit HPET mode, both Main_Counter_rtc_L and Main_Counter_rtc_H are valid.

**HPETx000001E8 (FCH::TMR::HPET::nxttmrremain\_l)**

Read-only. Reset: FFFF\_FFFFh.

\_aliasHOST; HPETx000001E8; HPET=FED0\_0000h

Bits	Description
31:0	<b>nxttmrremain_l.</b> Read-only. Reset: FFFF_FFFFh. <b>Description:</b> This register tells SW how many timer ticks remain before the next enabled comparator interrupt. For 32-bit HPET mode, only NxtTmrRemain_L is valid. For 64-bit HPET mode, both NxtTmrRemain_L and NxtTmrRemain_H are valid.

**HPETx000001EC (FCH::TMR::HPET::nxttmrremain\_h)**

Read-only. Reset: FFFF\_FFFFh.

\_aliasHOST; HPETx000001EC; HPET=FED0\_0000h

Bits	Description
31:0	<b>nxttmrremain_h.</b> Read-only. Reset: FFFF_FFFFh. <b>Description:</b> This register tells SW how many timer ticks remain before the next enabled comparator interrupt. For 32-bit HPET mode, only NxtTmrRemain_L is valid. For 64-bit HPET mode, both NxtTmrRemain_L and NxtTmrRemain_H are valid.

**7.3.5 Watchdog Timer (WDT) Registers**

Watchdog timer registers are accessed through two methods:

- Memory access to Watchdog Timer memory address range from FEB0\_0000h to FEB0\_000Fh. Program FCH::PM::decodeen[watchdogtmren]=1 to enable Watchdog Timer decoding.
- Memory mapped access through the AcpiMmio region. The Watchdog Timer registers start from FED8\_0000h+B00h. See FCH::PM::isacontrol[mmioen].

WDTx00000000 (FCH::TMR::WDT::watchdogcontrol)	
Read-write.	
_aliasHOST; WDTx00000000; WDT=FED8_0B00h	
Bits	Description
31:8	Reserved.
7	<b>watchdogtrigger_wo.</b> Read-write. Reset: 0. <b>Description:</b> Write only. Setting this bit triggers the watchdog to start a new count interval, counting down from the value that was last written to the Watchdog Count Register. This bit is always read as zero. Setting this bit has no effect if the watchdog is disabled or stopped.
6:5	Reserved.
4	<b>watchdogactionen.</b> Read-write. Reset: 1. <b>Description:</b> This bit is reset when enter S0I3/Zstate. It can only be set after system enter S0I3/Zstate and recover back. It is set by SMU. When this bit is high, the actions in bit2 can be implemented. When this bit is low, although WatchDogFired is set, the actions will still be suppressed.
3	<b>watchdogdisable.</b> Read-write. Reset: 1. <b>Description:</b> This bit reflects the state of PMIO_Reg:00[7]. Writing to this bit has no effect. 0: Enable 1: Disable
2	<b>watchdogaction.</b> Read-write. Reset: 0. <b>Description:</b> This bit determines the action to be taken when the watchdog timer expires. 0: System reset 1: System power off The bit is only valid when the watchdog is enabled.
1	<b>watchdogfired.</b> Read-write. Reset: 0. A value of "1" indicates that the watchdog timer has expired and caused the current restart. The bit is cleared by writing a "1" to bit 1 in the Watchdog Control register. Writing a "0" has no effect. The bit is cleared by a power cycle or by the operating system and it must remain cleared for any restart that is not caused by the watchdog timer firing. The bit is only valid when the watchdog is enabled.
0	<b>watchdogrunstopb.</b> Read-write. Reset: 0. <b>Description:</b> This bit is used to control or indicate whether the watchdog is in the Running and Stopped states. 1: Watchdog is in the Running state 0: Watchdog is in the Stopped state If the watchdog is in the Stopped state and a 1 is written to bit [0], the watchdog moves to the Running state, but a count interval is not started until a 1 is written to bit [7]. If the watchdog is in the Running state, writing a 1 to bit 0 has no effect. The bit is only valid when the watchdog is enabled.

WDTx00000004 (FCH::TMR::WDT::watchdogcount)	
Read-write.	
_aliasHOST; WDTx00000004; WDT=FED8_0B00h	
Bits	Description
31:16	Reserved.
15:0	<b>watchdogcount.</b> Read-write. Reset: XXXXXXXXXXXXXXXXb. <b>Description:</b> This defines the countdown time for the counter. The units are defined in the Units field in the Watchdog Resource Table (WDRT). The maximum value is defined in the Max Count field in the WDRT. Reading this register returns in the current counter value.

### 7.3.6 Wake Alarm Device (AcDcTimer) Registers

In Family 1Ah Models , signal iBatteryModeB is not tied off, it connects to BP\_AC\_PRESENCE or BP\_AGPIOWAKE.



The AC/DC timer registers are used to control the wake alarm device. They are accessed through the AcpiMmio region. The AC DC timer registers range from FED8\_0000h+1D00h to FED8\_0000h+1DFFh.

#### ACDCx00000000 (FCH::TMR::ACDC::ac\_timer\_value)

Read-write. Reset: FFFF\_FFFFh.

\_aliasHOST; ACDCx00000000; ACDC=FED8\_1D00h

Bits	Description
31:0	<p><b>ac_timer_value.</b> Read-write. Reset: FFFF_FFFFh.</p> <p><b>Description:</b> Writing the register to a value other than FFFFFFFFh will start the AC timer:  Reading this register returns the current value of AC timer.  When AC or DC Timer generate wake up event, this register will be reset to FFFFFFFFh by hardware.  FFFFFFFFh: Disable AC timer.  FFFFFFFEh ~ 00000001h: The value indicates the number of seconds between the time when the AC timer is programmed and the time when it expires.  00000000h: The AC timer will wake up the system instantly.</p>

#### ACDCx00000004 (FCH::TMR::ACDC::ac\_expired\_timer\_policy)

Read-write. Reset: FFFF\_FFFFh.

\_aliasHOST; ACDCx00000004; ACDC=FED8\_1D00h

Bits	Description
31:0	<p><b>ac_expired_timer_policy.</b> Read-write. Reset: FFFF_FFFFh.</p> <p><b>Description:</b> If AC timer expired when current power source is DC, the wake signal won't be asserted. If the power source is switched back to AC when the AC timer is already expired, we will wait for the number of seconds defined in this register and then wake up the system.  When AC or DC Timer generate wake up event, this register will be reset to FFFFFFFFh by hardware.  FFFFFFFFh: Disable AC expired timer policy.  FFFFFFFEh ~ 00000001h: The value indicates the number of seconds between the time when the power is switched to AC and the time when it generates the wake-up event.  00000000h: The expired AC timer will wake up the system instantly once the power source is switched to AC.</p>

#### ACDCx00000008 (FCH::TMR::ACDC::ac\_timer\_status)

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ACDCx00000008; ACDC=FED8\_1D00h

Bits	Description
31:2	Reserved.
1	<p><b>ac_timer_wakeup.</b> Read-write. Reset: 0.</p> <p><b>Description:</b> 1: Wake-up was caused by AC timer expiration  0: Wake-up was not caused by AC timer expiration  Write 1 clear</p>
0	<p><b>ac_timer_expired.</b> Read-write. Reset: 0.</p> <p><b>Description:</b> 1: AC timer expired  0: AC timer not expired  Write 1 clear</p>



**ACDCx00000010 (FCH::TMR::ACDC::dc\_timer\_value)**

Read-write. Reset: FFFF\_FFFFh.

\_aliasHOST; ACDCx00000010; ACDC=FED8\_1D00h

Bits	Description
31:0	<p><b>dc_timer_value.</b> Read-write. Reset: FFFF_FFFFh.</p> <p><b>Description:</b> Writing the register to a value other than FFFFFFFFh will start the DC timer: Reading this register returns the current value of DC timer. When AC or DC Timer generate wake up event, this register will be reset to FFFFFFFFh by hardware. FFFFFFFFh: Disable DC timer. FFFFFFFFEh ~ 00000001h: The value indicates the number of seconds between the time when the DC timer is programmed and the time when it expires. 00000000h: The DC timer will wake up the system instantly.</p>

**ACDCx00000014 (FCH::TMR::ACDC::dc\_expired\_timer\_policy)**

Read-write. Reset: FFFF\_FFFFh.

\_aliasHOST; ACDCx00000014; ACDC=FED8\_1D00h

Bits	Description
31:0	<p><b>dc_expired_timer_policy.</b> Read-write. Reset: FFFF_FFFFh.</p> <p><b>Description:</b> If DC timer expired when current power source is AC, the wake signal won't be asserted. If the power source is switched back to DC when the DC timer is already expired, we will wait for the number of seconds defined in this register and then wake up the system. When AC or DC Timer generate wake up event, this register will be reset to FFFFFFFFh by hardware. FFFFFFFFh: Disable DC expired timer policy. FFFFFFFFEh ~ 00000001h: The value indicates the number of seconds between the time when the power is switched to DC and the time when it generates the wake-up event. 00000000h: The expired DC timer will wake up the system instantly once the power source is switched to DC.</p>

**ACDCx00000018 (FCH::TMR::ACDC::dc\_timer\_status)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ACDCx00000018; ACDC=FED8\_1D00h

Bits	Description
31:2	Reserved.
1	<p><b>dc_timer_wakeup.</b> Read-write. Reset: 0.</p> <p><b>Description:</b> 1: Wake-up was caused by DC timer expiration 0: Wake-up was not caused by DC timer expiration Write 1 clear</p>
0	<p><b>dc_timer_expired.</b> Read-write. Reset: 0.</p> <p><b>Description:</b> 1: DC timer expired 0: DC timer not expired Write 1 clear</p>

**ACDCx00000020 (FCH::TMR::ACDC::acdctimer\_ctrl)**

Read-write. Reset: 0000\_1000h.

\_aliasHOST; ACDCx00000020; ACDC=FED8\_1D00h

Bits	Description
31:13	Reserved.
12	<b>no_event_in_g0</b> . Read-write. Reset: 1. Debug purpose. Please don't change it.
11:10	<b>sel_wake_rst_1_0</b> . Read-write. Reset: 0h. Debug purpose. Please don't change it.
9	<b>dc_timer_event_en</b> . Read-write. Reset: 0. <b>Description:</b> 1: Disable AC Timer to wake up system 0: Disable
8	<b>ac_timer_event_en</b> . Read-write. Reset: 0. <b>Description:</b> 1: Enable AC Timer to wake up system 0: Disable
7:1	Reserved.
0	<b>busy</b> . Read-write. Reset: 0. <b>Description:</b> Read only. Right after AC_TIMER_VALUE or DC_TIMER_VALUE is programmed, the hardware will set Busy bit to 1. The hardware will clear Busy bit once the programming is done and the corresponding timer is started properly. Before the software write the AC_TIMER_VALUE or DC_TIMER_VALUE registers, it has to read the Busy bit and make sure it is 0, otherwise, the hardware will just ignore the programming action from software. For the registers other than AC_TIMER_VALUE and DC_TIMER_VALUE, there is no such limitation.

**7.3.7 Always On Always Connected (AOAC) Registers***Table 100: FCH Device D3 control/status mapping*

Controller Block	FCH Power Group	D3 Control Register	D3 Status Register	ClkOk (From block to ACPI)	AOACCtrl (From ACPI to block)
Clk Gen	PG1	DevCtrl 0	DevSts 0	Tied high	Connected PllRstB and PllLock
AB	PG1	DevCtrl 1	DevSts 1	Tied high	Connected
ACPI S0 (fch_acpismbus)	PG1	DevCtrl 2	DevSts 2	Tied high	Connected
ACPI S5 (fch_acpi_s5)	S5	DevCtrl 3	DevSts 3	Tied high	Not used
LPC	PG1	DevCtrl 4	DevSts 4	Tied high	Connected
I2C0	PG1a	DevCtrl 5	DevSts 5	Tied high	Connected
I2C1	PG2	DevCtrl 6	DevSts 6	Tied high	Connected
I2C2	PG2	DevCtrl 7	DevSts 7	Tied high	Connected
I2C3	PG2	DevCtrl 8	DevSts 8	Tied high	Connected
UART0	PG2	DevCtrl 11	DevSts 11	Tied high	Connected
UART1	PG2	DevCtrl 12	DevSts 12	Tied high	Connected
I3C1	PG2	DevCtrl 13	DevSts 13	Tied high	Connected
I3C2	PG2	DevCtrl 14	DevSts 14	Tied high	Connected
I3C3	PG2	DevCtrl 15	DevSts 15	Tied high	Connected
UART2	PG2	DevCtrl 16	DevSts 16	Tied high	Connected
AMBA	PG1	DevCtrl 17	DevSts 17	Tied high	Connected

UART4	PG2	DevCtrl 20	DevSts 20	Tied high	Connected
I3C0	PG1a	DevCtrl 21	DevSts 21	Tied high	Connected
UART3	PG2	DevCtrl 26	DevSts 26	Tied high	Connected
eSPI	PG1	DevCtrl 27	DevSts 27	Tied high	Connected
APU	PG1	DevCtrl 31	DevSts 31	Connected	Conneced FCH pin name LDT_PWROK and LDT_RST_L

**AOACx00000000 (FCH::AOAC::perfmon\_control)**

Read-write. Reset: 0000\_0000h.

**NOTE 1:**

Each block's busy signal has a traffic counter. The traffic counter keeps track of the amount of traffic in a specified timer interval. Each time interval consists of 65280 time slots. We can change the length of the time slot and time interval by changing the MonPeriodSel[1:0] register. If the busy signal from a block has asserted during a time slot, the counter will increase by 1 no matter how long the busy signal was during that time slot. At the end of the time interval, the counter value will be scaled by a weight and updated to the output of the traffic counter.

Each block has a traffic counter and software can read the weighted counter values of each block (TrafficCount\_00 ~ 07). Hardware also provides a sum of those weighted counter values through register TrafficCount\_All.

\_aliasHOST; AOACx00000000; AOAC=FED8\_1E00h

Bits	Description
31:30	Reserved.
29	<b>trafficsts.</b> Read-write. Reset: 0. <b>Description:</b> Write-1-clear 1: There are less (TrafficLess=1) or more (TrafficLess=0) traffic than the threshold defined in BusyTimeThreshold. 0: There are more (TrafficLess=1) or less (TrafficLess=0) traffic than the threshold defined in BusyTimeThreshold.
28	<b>intrsts.</b> Read-write. Reset: 0. <b>Description:</b> Write-1-clear 1: There are less (IntrLess=1) or more (IntrLess=0) traffic than the threshold defined in IntrTimeThreshold. 0: There are more (IntrLess=1) or less (IntrLess=0) traffic than the threshold defined in IntrTimeThreshold.
27	<b>cnt1source.</b> Read-write. Reset: 0. <b>Description:</b> 1: Assign traffic_cnt01 to Sata traffic. 0: Assign traffic_cnt01 to Sata Port 1 active (not in partial/slumber).
26	<b>cnt0source.</b> Read-write. Reset: 0. <b>Description:</b> 1: Assign traffic_cnt00 to "AnyBusy". 0: Assign traffic_cnt00 to Sata Port 0 active (not in partial/slumber). (See NOTE 2)
25	<b>checkinterrupt.</b> Read-write. Reset: 0. Enable interrupt counter (IntrCount)
24	<b>checkc3.</b> Read-write. Reset: 0. <b>Description:</b> 1: Check C3 state when counting "AnyBusy" in traffic_cnt00. 0: Don't check C3. (See NOTE 2)
23:8	Reserved.
7:6	<b>monperiodsel_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Time slot = 15 ns Time interval = 979.2 us 01: Time slot = 240 ns Time interval = 15.667 ms 10: Time slot = 1.92 us Time interval = 125.34 ms 11: Time slot = 15.36 us Time interval = 1 s (See NOTE 1)
5	Reserved.
4	<b>intrless.</b> Read-write. Reset: 0. <b>Description:</b> 1: Generate interrupt status (IntrSts) when there is less interrupt than the specified threshold (IntrTimeThreshold) 0: Generate interrupt status (IntrSts) when there is more traffic than the specified threshold (IntrTimeThreshold)
3	<b>trafficless.</b> Read-write. Reset: 0. <b>Description:</b> 1: Generate traffic status (TrafficSts) when there is less traffic than the specified threshold (BusyTimeThreshold) 0: Generate traffic status (TrafficSts) when there is more traffic than the specified threshold (BusyTimeThreshold)

2	<b>interruptscien.</b> Read-write. Reset: 0. <b>Description:</b> 1: Generate SCI when there is interrupt status (IntrSts) 0: Disable
1	<b>trafficscien.</b> Read-write. Reset: 0. <b>Description:</b> 1: Generate SCI when there is traffic status (TrafficSts) 0: Disable
0	<b>perfmonenable.</b> Read-write. Reset: 0. Global enable of Performance Monitor

**AOACx00000004 (FCH::AOAC::perfmon\_time\_limit)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; AOACx00000004; AOAC=FED8\_1E00h

Bits	Description
31:16	<b>intrtimethreshold.</b> Read-write. Reset: 0000h. Specify the counter threshold for generating Interrupt SCI.
15:0	<b>busytimethreshold.</b> Read-write. Reset: 0000h. Specify the counter threshold for generating Traffic SCI.

**AOACx00000008 (FCH::AOAC::perfmon\_weight\_3\_2\_1\_0)**

Reset: 0000\_0000h.

\_aliasHOST; AOACx00000008; AOAC=FED8\_1E00h

Bits	Description
31:0	Reserved.

**AOACx0000000C (FCH::AOAC::perfmon\_weight\_7\_6\_5\_4)**

Reset: 0000\_0000h.

\_aliasHOST; AOACx0000000C; AOAC=FED8\_1E00h

Bits	Description
31:0	Reserved.

**AOACx00000010 (FCH::AOAC::perfmon\_traf\_cnt\_1\_0)**

Reset: 0000\_0000h.

\_aliasHOST; AOACx00000010; AOAC=FED8\_1E00h

Bits	Description
31:0	Reserved.

**AOACx00000014 (FCH::AOAC::perfmon\_traf\_cnt\_3\_2)**

Reset: 0000\_0000h.

\_aliasHOST; AOACx00000014; AOAC=FED8\_1E00h

Bits	Description
31:0	Reserved.

**AOACx00000018 (FCH::AOAC::perfmon\_traf\_cnt\_5\_4)**

Reset: 0000\_0000h.

\_aliasHOST; AOACx00000018; AOAC=FED8\_1E00h

Bits	Description
31:0	Reserved.

**AOACx0000001C (FCH::AOAC::perfmon\_traf\_cnt\_7\_6)**

Reset: 0000\_0000h.

\_aliasHOST; AOACx0000001C; AOAC=FED8\_1E00h

Bits	Description
31:0	Reserved.

**AOACx00000020 (FCH::AOAC::perfmon\_traf\_cnt\_all)**

Read-only. Reset: 0000\_0000h.

TrafficCount\_All[26:0] = Internal-Weighted-Count00 [23:0] + + Internal-Weighted-Count07 [23:0]

\_aliasHOST; AOACx00000020; AOAC=FED8\_1E00h

Bits	Description
31:27	Reserved.
26:0	<b>trafficcount_all_26_0</b> . Read-only. Reset: 000_0000h. Sum of weighted counts from each traffic_cntXX blocks.

**AOACx00000024 (FCH::AOAC::perfmon\_intr\_cnt)**

Read-only. Reset: 0000\_0000h.

\_aliasHOST; AOACx00000024; AOAC=FED8\_1E00h

Bits	Description
31:24	<b>swbusy</b> . Read-only. Reset: 00h.
23:16	<b>swbusyen</b> . Read-only. Reset: 00h.
15:0	<b>intrcount_15_0</b> . Read-only. Reset: 0000h.

**AOACx00000030 (FCH::AOAC::althpet\_timer\_l)**

Read-only.

Register 30h and 34h are read-only. It shows the current time with higher resolution than traditional RTC timer. AltMmTimer field shows the time with 64KHz resolution. It will wrap around at every second.

\_aliasHOST; AOACx00000030; AOAC=FED8\_1E00h

Bits	Description
31:24	<b>minute_7_0</b> . Read-only. Reset: XXXXXXXXXb. Minute field of the ALTHPET_TIMER
23:16	<b>second_7_0</b> . Read-only. Reset: XXXXXXXXXb. Second field of the ALTHPET_TIMER
15:0	<b>altmmtimer</b> . Read-only. Reset: XXXXXXXXXXXXXXXXXb. AltMmTimer field of the ALTHPET_TIMER

**AOACx00000034 (FCH::AOAC::althpet\_timer\_h)**

Read-only.

\_aliasHOST; AOACx00000034; AOAC=FED8\_1E00h

Bits	Description
31:24	<b>year_7_0</b> . Read-only. Reset: XXXXXXXXXb. Year field of the ALTHPET_TIMER
23:16	<b>month_7_0</b> . Read-only. Reset: XXXXXXXXXb. Month field of the ALTHPET_TIMER
15:8	<b>day_7_0</b> . Read-only. Reset: XXXXXXXXXb. Day field of the ALTHPET_TIMER
7:0	<b>hour_7_0</b> . Read-only. Reset: XXXXXXXXXb. Hour field of the ALTHPET_TIMER

**AOACx00000038 (FCH::AOAC::althpet\_alarm\_l)**

Read-write. Reset: 0000\_0000h.

AltMmTimer can be programmed to trigger SMI (Event 43).

Whenever ALTHPET\_TIMER (L and H) matches the value defined in ALTHPET\_ALARM (L and H), we will generate a pulse on AltMmTimerSts (Event 43).

\_aliasHOST; AOACx00000038; AOAC=FED8\_1E00h

Bits	Description
31	Reserved.
30:24	<b>minute_alarm_6_0</b> . Read-write. Reset: 00h. Minute field of the ALTHPET_ALARM
23	Reserved.
22:16	<b>second_alarm_6_0</b> . Read-write. Reset: 00h. Second field of the ALTHPET_ALARM
15:0	<b>altmmtimer_alarm</b> . Read-write. Reset: 0000h. AltMmTimer field of the ALTHPET_ALARM

**AOACx0000003C (FCH::AOAC::althpet\_alarm\_h)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; AOACx0000003C; AOAC=FED8\_1E00h

Bits	Description
31:24	<b>year_alarm_7_0.</b> Read-write. Reset: 00h. Year field of the ALTHPET_ALARM
23:21	Reserved.
20:16	<b>month_alarm_4_0.</b> Read-write. Reset: 00h. Month field of the ALTHPET_ALARM
15:14	Reserved.
13:8	<b>day_alarm_5_0.</b> Read-write. Reset: 00h. Day field of the ALTHPET_ALARM
7:6	Reserved.
5:0	<b>hour_alarm_5_0.</b> Read-write. Reset: 00h. Hour field of the ALTHPET_ALARM

**AOACx00000040 (FCH::AOAC::devctrl\_0)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx00000040; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol.</b> Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb.</b> Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok.</b> Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb.</b> Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev.</b> Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate.</b> Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate.</b> Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx00000041 (FCH::AOAC::devsts\_0)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000041; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000042 (FCH::AOAC::devctrl\_1)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx00000042; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold



**AOACx00000043 (FCH::AOAC::devsts\_1)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000043; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000044 (FCH::AOAC::devctrl\_2)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx00000044; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx00000045 (FCH::AOAC::devsts\_2)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000045; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000046 (FCH::AOAC::devctrl\_3)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx00000046; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx00000047 (FCH::AOAC::devsts\_3)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000047; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrestb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000048 (FCH::AOAC::devctrl\_4)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx00000048; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx00000049 (FCH::AOAC::devsts\_4)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000049; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx0000004A (FCH::AOAC::devctrl\_5)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx0000004A; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx0000004B (FCH::AOAC::devsts\_5)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx0000004B; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx0000004C (FCH::AOAC::devctrl\_6)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx0000004C; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx0000004D (FCH::AOAC::devsts\_6)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx0000004D; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx0000004E (FCH::AOAC::devctrl\_7)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx0000004E; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx0000004F (FCH::AOAC::devsts\_7)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx0000004F; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000050 (FCH::AOAC::devctrl\_8)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx00000050; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold



**AOACx00000051 (FCH::AOAC::devsts\_8)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000051; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000052 (FCH::AOAC::devctrl\_9)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx00000052; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold



**AOACx00000053 (FCH::AOAC::devsts\_9)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000053; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000054 (FCH::AOAC::devctrl\_10)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx00000054; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx00000055 (FCH::AOAC::devsts\_10)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000055; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000056 (FCH::AOAC::devctrl\_11)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx00000056; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx00000057 (FCH::AOAC::devsts\_11)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000057; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000058 (FCH::AOAC::devctrl\_12)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx00000058; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx00000059 (FCH::AOAC::devsts\_12)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000059; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx0000005A (FCH::AOAC::devctrl\_13)**

Read-write. Reset: 74h.

\_aliasHOST; AOACx0000005A; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 0. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 0h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx0000005B (FCH::AOAC::devsts\_13)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx0000005B; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx0000005C (FCH::AOAC::devctrl\_14)**

Read-write. Reset: 74h.

\_aliasHOST; AOACx0000005C; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 0. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 0h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx0000005D (FCH::AOAC::devsts\_14)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx0000005D; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx0000005E (FCH::AOAC::devctrl\_15)**

Read-write. Reset: 74h.

\_aliasHOST; AOACx0000005E; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 0. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 0h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx0000005F (FCH::AOAC::devsts\_15)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx0000005F; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000060 (FCH::AOAC::devctrl\_16)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx00000060; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold



**AOACx00000061 (FCH::AOAC::devsts\_16)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000061; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000062 (FCH::AOAC::devctrl\_17)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx00000062; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold



**AOACx00000063 (FCH::AOAC::devsts\_17)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000063; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000064 (FCH::AOAC::devctrl\_18)**

Read-write. Reset: 74h.

\_aliasHOST; AOACx00000064; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 0. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 0h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx00000065 (FCH::AOAC::devsts\_18)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000065; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000066 (FCH::AOAC::devctrl\_19)**

Read-write. Reset: 74h.

\_aliasHOST; AOACx00000066; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 0. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 0h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx00000067 (FCH::AOAC::devsts\_19)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000067; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000068 (FCH::AOAC::devctrl\_20)**

Read-write. Reset: 74h.

\_aliasHOST; AOACx00000068; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 0. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 0h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx00000069 (FCH::AOAC::devsts\_20)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000069; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx0000006A (FCH::AOAC::devctrl\_21)**

Read-write. Reset: 74h.

\_aliasHOST; AOACx0000006A; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 0. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 0h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx0000006B (FCH::AOAC::devsts\_21)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx0000006B; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx0000006C (FCH::AOAC::devctrl\_22)**

Read-write. Reset: 74h.

\_aliasHOST; AOACx0000006C; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 0. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 0h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx0000006D (FCH::AOAC::devsts\_22)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx0000006D; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx0000006E (FCH::AOAC::devctrl\_23)**

Read-write. Reset: 74h.

\_aliasHOST; AOACx0000006E; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 0. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 0h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx0000006F (FCH::AOAC::devsts\_23)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx0000006F; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000070 (FCH::AOAC::devctrl\_24)**

Read-write. Reset: 74h.

\_aliasHOST; AOACx00000070; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwrndev</b> . Read-write. Reset: 0. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 0h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold



**AOACx00000071 (FCH::AOAC::devsts\_24)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000071; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000072 (FCH::AOAC::devctrl\_25)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx00000072; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold



**AOACx00000073 (FCH::AOAC::devsts\_25)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000073; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000074 (FCH::AOAC::devctrl\_26)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx00000074; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx00000075 (FCH::AOAC::devsts\_26)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000075; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000076 (FCH::AOAC::devctrl\_27)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx00000076; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx00000077 (FCH::AOAC::devsts\_27)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000077; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwr_rstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000078 (FCH::AOAC::devctrl\_28)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx00000078; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx00000079 (FCH::AOAC::devsts\_28)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx00000079; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx0000007A (FCH::AOAC::devctrl\_29)**

Read-write. Reset: 74h.

\_aliasHOST; AOACx0000007A; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 0. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 0h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx0000007B (FCH::AOAC::devsts\_29)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx0000007B; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx0000007C (FCH::AOAC::devctrl\_30)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx0000007C; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx0000007D (FCH::AOAC::devsts\_30)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx0000007D; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrstb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx0000007E (FCH::AOAC::devctrl\_31)**

Read-write. Reset: 7Dh.

\_aliasHOST; AOACx0000007E; AOAC=FED8\_1E00h

Bits	Description
7	<b>isswcontrol</b> . Read-write. Reset: 0. <b>Description:</b> 1: Sw controls control signals (PwrRstB, RefClkOk, RstB) to the device. 0: Hw controls control signals (PwrRstB, RefClkOk, RstB) to the device. To avoid glitch on the reset signals, SW must set IsSwControl bit first and then program SwPwrOnRstB, SwRefClkOk and SwRstB.
6	<b>swrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: RstB is asserted to the device if IsSwControl is set to 1. 1: RstB is de-asserted to the device if IsSwControl is set to 1.
5	<b>swrefclkok</b> . Read-write. Reset: 1. <b>Description:</b> 0: RefClkOk is de-asserted to the device if IsSwControl is set to 1. 1: RefClkOk is asserted to the device if IsSwControl is set to 1.
4	<b>swpwrnrstb</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRstB is asserted to the device if IsSwControl is set to 1. 1: PwrRstB is de-asserted to the device if IsSwControl is set to 1.
3	<b>pwronddev</b> . Read-write. Reset: 1. <b>Description:</b> If SwControl is set to 0, SW can write this bit to trigger a HW controlled reset sequence to the device. 0: put device in reset state. 1: put device out of reset state.
2	<b>devicestate</b> . Read-write. Reset: 1. <b>Description:</b> SW records the device power state in this field. 0: device power is removed 1: device power is applied
1:0	<b>targeteddevicestate</b> . Read-write. Reset: 1h. <b>Description:</b> SW records D-State of the device in this field. 00: D0 UnInitialized 01: D0 Initialized 10: D1/D2/D3 Hot 11: D3 Cold

**AOACx0000007F (FCH::AOAC::devsts\_31)**

Read-only. Reset: 27h.

\_aliasHOST; AOACx0000007F; AOAC=FED8\_1E00h

Bits	Description
7	<b>stat1</b> . Read-only. Reset: 0. State of device.
6	<b>stat0</b> . Read-only. Reset: 0. State of device.
5	<b>clkok_state</b> . Read-only. Reset: 1. State of device's ClkOk signal.
4	<b>d3cold</b> . Read-only. Reset: 0. State of device's D3Cold signal.
3	<b>devoffgating_state</b> . Read-only. Reset: 0. State of device's DevOffGating signal.
2	<b>rstb_state</b> . Read-only. Reset: 1. State of device's RstB signal.
1	<b>refclkok_state</b> . Read-only. Reset: 1. State of device's RefClkOk signal.
0	<b>pwrrestb_state</b> . Read-only. Reset: 1. State of device's PwrRstB signal.

**AOACx00000080 (FCH::AOAC::shadow\_reg\_request)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; AOACx00000080; AOAC=FED8\_1E00h

Bits	Description
31	<b>shadow_rst</b> . Read-write. Reset: 0. <b>Description:</b> SW can write this bit to reset shadow register control logics. (Write 1 to reset, and then write 0 to release the reset) Note: SHADOW_hang means somehow the current request is not completed. SW should do a SHADOW_rst and request again. The Shadow Reg Status bit should remain previous state from last successful save/restore of the corresponding IP. So we don't want to reset Shadow Reg Status bit by SHADOW_rst.
30	<b>request_sel</b> . Read-write. Reset: 0. <b>Description:</b> SW write this bit to select Request bank. 0: Select StoreRequest 1: Select RestoreRequest
29:0	<b>request</b> . Read-write. Reset: 0000_0000h. <b>Description:</b> Request[29:0] are actually shared by two register banks: StoreRequest[29:0] and RestoreRequest[29:0]. SW write Request_Sel bit to choose which bank to access. SW write 1 to these bits to request store or restore. The request type depends on which bank the bit is located (StoreRequest or RestoreRequest). Each bit is corresponding to a device being stored/restored. Write 0 has no effect to these registers. Once store/restore is done for a device, the corresponding bit will be cleared to 0 by HW. Only HW will be able to clear these bits.

**AOACx00000084 (FCH::AOAC::shadow\_reg\_status)**

Read-write. Reset: 3FFF\_FFFFh.

\_aliasHOST; AOACx00000084; AOAC=FED8\_1E00h

Bits	Description
31:30	Reserved.
29:0	<b>status</b> . Read-write. Reset: 3FFF_FFFFh. <b>Description:</b> Reading these bits returns the status of the store/restore action. Each bit represent the status of a device. 0 means shadow register logic has finished store action for a device. 1 means shadow register has finished restore action for a device. SW can also write this register to 1 or 0. This provide SW a way to override the status.



**AOACx00000088 (FCH::AOAC::shadow\_reg\_sram\_addr)**

Read-write. Reset: 0000\_0000h.

SW has indirect access to the SRAM in shadow register block via ShadowSRAM\_Addr and ShadowSRAM\_Data. SW can write ShadowSRAM\_Addr to an address it wants to access, and then read or write ShadowSRAM\_Data.

\_aliasHOST; AOACx00000088; AOAC=FED8\_1E00h

Bits	Description
31:24	<b>century_7_0</b> . Read-write. Reset: 00h. Century info from BcdTime. It's read only, write to it has no effect.
23:16	Reserved.
15:0	<b>shadowsram_addr</b> . Read-write. Reset: 0000h. Shadow SRAM address (see NOTE)

**AOACx0000008C (FCH::AOAC::shadow\_reg\_sram\_data)**

Read-write. Reset: 0000\_0000h.

SW has indirect access to the SRAM in shadow register block via ShadowSRAM\_Addr and ShadowSRAM\_Data. SW can write ShadowSRAM\_Addr to an address it wants to access, and then read or write ShadowSRAM\_Data.

\_aliasHOST; AOACx0000008C; AOAC=FED8\_1E00h

Bits	Description
31:0	<b>shadowsram_data</b> . Read-write. Reset: 0000_0000h. Shadow SRAM data (see NOTE)

**AOACx00000090 (FCH::AOAC::shadow\_reg\_hw\_init\_en)**

\_aliasHOST; AOACx00000090; AOAC=FED8\_1E00h

Bits	Description
31	<b>shadowsram_inds</b> . Read-only. Reset: X. <b>Description:</b> Deep Sleep status of Shadow SRAM. 0: Shadow SRAM is in normal function mode 1: Shadow SRAM is in Deep Sleep mode
30	<b>forcemasterack</b> . Read-write. Reset: 0. <b>Description:</b> 0: Shadow Master dynamically turn on/off its internal clock base on the requests. 1: Shadow Master internal clock always enabled.
29:0	<b>hwinit_en</b> . Read-write. Reset: 0820_E016h. These bits configure the capability of "hardware initiated store/restore" for each device. When a device's HwInit_En bit is enabled, hardware will automatically initiate a store for that device before going into S0i3 state, and a restore when waking up from S0i3 state.



**AOACx00000094 (FCH::AOAC::s0i3ctrl)**

Reset: 0000\_0000h.

\_aliasHOST; AOACx00000094; AOAC=FED8\_1E00h

Bits	Description
31:29	Reserved.
28	<b>memrepairdonemask_chk_bit.</b> Read-write. Reset: 0. MemRepairDoneMask_chk_bit
27	<b>en_pcirstwake_chk_bit.</b> Read-write. Reset: 0. <b>Description:</b> Enable PciRstB gen FCH_ZSC_wake during Z8 1'b0: disable 1'b1: enable
26	<b>en_s0wake_chk_bit.</b> Read-write. Reset: 0. <b>Description:</b> Enable s0 wake event for Z8 1'b0: disable 1'b1: enable
25	<b>fchz8support.</b> Read-write. Reset: 0. Enable FCH Z8 support
24	<b>fchzstateenable.</b> Read-write. Reset: 0. Enable FCH Z-state support
23	<b>selectivermw.</b> Read-write. Reset: 0. Selectively turn on read-modify-write for shadow table entries with MSB set to 1, after read-modify-write is turned off globally with RestoreRmwDisable=1
22	<b>ignorebitmap.</b> Read-write. Reset: 0. Ignore bitmap field in shadow table entry and save/restore all register bits
21	<b>restorermwdisable.</b> Read-write. Reset: 0. Replace read-modify-write with write during restore
20	<b>localshadowtable.</b> Read-write. Reset: 0. Load save/restore information from local shadow table instead of each IP
19	Reserved.
18	<b>s0i20_req.</b> Write-only. Reset: 0. <b>Description:</b> Read always returns 0. SMU can write 1 to this bit to request FCH to Slow Down clock (S0I20) when PM_regx50[0] =1
17	<b>apuplloffreq.</b> Write-only. Reset: 0. <b>Description:</b> Read always returns 0. SMU can write 1 to this bit to request FCH to shutdown CGPLL (S0I21) when PM_regx50[0] =1
16	<b>intrblocked.</b> Read-only. Reset: 0. <b>Description:</b> If FCH is sending an interrupt message when SW or SMU write 1 to InterruptDis, the interrupt will be blocked "gracefully". We will wait until the interrupt is sent and then block the interrupt. This bit indicates the interrupt blocking status. 0: FCH interrupts are not blocked 1: FCH interrupts are blocked
15	<b>interruptdis.</b> Read-write. Reset: 0. <b>Description:</b> SW is able to set and clear this bit. HW can only set this bit. 1: interrupt from FCH is disabled. 0: interrupt from FCH is enabled.
14	<b>arbiterdis.</b> Read-write. Reset: 0. <b>Description:</b> SW is able to set and clear this bit. HW can only set this bit. 1: FCH upstream arbiter is disabled. 0: FCH upstream arbiter is enabled.
13	<b>maskioapic.</b> Read-write. Reset: 0. <b>Description:</b> 0: IOAPIC mask bits function normally 1: Mask all IOAPIC interrupts This bit is OR'd with all IOAPIC interrupt mask bits.
12	Reserved.
11	<b>anyupwakes0i3en.</b> Read-write. Reset: 0.

	<b>Description:</b> 0: S0i3 state can be wake up by GPIO controller and Shadow Timer. 1: S0i3 state can be wake up by GPIO controller, Shadow Timer and any pending upstream message in ACPI.
10	<b>shadowtimerwakests.</b> Read-only. Reset: 0. <b>Description:</b> HW will clear this bit during S0i3 entry. 1: Shadow Timer fired and caused wake-up from S0i3 state. 0: Shadow Timer did not fire.
9	<b>s0a3resume.</b> Read-write. Reset: 0. Set to 1 by S0i3 exit to indicate the system has resumed from S0i3 state. The bit is cleared by S0i3 entry or write-1-cleared by Sw.
8	<b>s0a3enter.</b> Read-only. Reset: 0. <b>Description:</b> 1: indicate the system is in S0i3 state. 0: indicate the system is in S0 state.
7:5	Reserved.
4	<b>s0a3trigger.</b> Write-only. Reset: 0. <b>Description:</b> Read always return 0. Setting the bit to 1 to trigger S0i3 power down sequence.
3	Reserved.
2	<b>s0a3ready2.</b> Read-write. Reset: 0. <b>Description:</b> Programmed by Driver. 0: S0i3 can't be entered 1: S0i3 can be entered
1	<b>s0a3onslps3b.</b> Read-write. Reset: 0. <b>Description:</b> 0: SLP_S3# doesn't assert in S0i3 state 1: SLP_S3# will assert in S0i3 state
0	<b>s0a3ready.</b> Read-write. Reset: 0. <b>Description:</b> Programmed by BIOS. 0: S0i3 can't be entered 1: S0i3 can be entered

**AOACx00000098 (FCH::AOAC::s0i3\_debug)**

Read-write.

\_aliasHOST; AOACx00000098; AOAC=FED8\_1E00h

Bits	Description
31:25	Reserved.
24	<b>zstatepwrGOODdisable.</b> Read-write. Reset: 0. Disable generating cold reset from PWRGOOD pin in Z-state
23:15	Reserved.
14	<b>s0i3_lpcrstmaskdisable.</b> Read-write. Reset: 0. Disable LPC reset masking during S0i3 and Z-state
13	<b>s0i3_dontwaitstpgnt.</b> Read-write. Reset: 1. <b>Description:</b> 0: S0i3 state machine will wait for STPGNT after it sens STPCLK 1: Don't wait
12	<b>s0i3_dontwaitsdb.</b> Read-write. Reset: 1. <b>Description:</b> 0: Wait for SDB (in PG1) to re-initialize before restore during S0i3 wake-up 1: Don't wait
11	<b>s0i3_dontblockpmiorst.</b> Read-write. Reset: 0. <b>Description:</b> 0: Block PciRstB to PMIO register block during S0i3 1: Do not block PciRstB to PMIO register block during S0i3
10	<b>s0i3_force_clk.</b> Read-write. Reset: 0. <b>Description:</b> 0: S0i3 state machine dynamically gate its local clock 1: S0i3 state machine local clock always on
9	<b>s0i3_bypassshdwreg.</b> Read-write. Reset: 0. <b>Description:</b> 0: During S0i3 entry and exit, use ShadowReg to store and restore context of PG1 devices. 1: Bypass ShadowReg
8	<b>s0i3_bypassshdwtmr.</b> Read-write. Reset: 0. <b>Description:</b> 0: During S0i3 entry and exit, use ShadowTimer to store and restore HPET and AcpiTimer. 1: Bypass ShadowTimer
7:0	<b>s0i3_xtal_settle_time.</b> Read-write. Reset: XXXXXXXXb. <b>Description:</b> Debug purpose. SW should leave it at default value.

**AOACx0000009C (FCH::AOAC::shadow\_timer\_ctrl)**

Read-write.

\_aliasHOST; AOACx0000009C; AOAC=FED8\_1E00h

Bits	Description
31:24	<b>restoredoffset64.</b> Read-write. Reset: 00h. A programmable offset to be added to restored HPET 64bit timer this is to accounted for restore time uncertainty.
23:16	<b>restoredoffset.</b> Read-write. Reset: 00h. A programmable offset to be added to restored HPET 32bit timer and ACPI timer this is to accounted for restore time uncertainty.
15:8	<b>earlycount.</b> Read-write. Reset: XXXXXXXXb. A programmable offset to be subtracted from the alarm value
7:3	Reserved.
2	<b>earlycountunit.</b> Read-write. Reset: 0. <b>Description:</b> 0: The unit of EarlyCount is 1 RTC clock period. 1: The unit of EarlyCount is 16 RTC clock period
1	<b>shadowacpitimeren.</b> Read-write. Reset: 0. Enable control to perform store/restore operation for ACPI timer
0	<b>shadowhpeten.</b> Read-write. Reset: 0. Enable control to perform store/restore operation from HPET timer

**AOACx000000A0 (FCH::AOAC::pwrgood\_ctrl)**

Read-write. Reset: EE00\_E03Bh.

NOTE1:

NOTE2:

\_aliasHOST; AOACx000000A0; AOAC=FED8\_1E00h

Bits	Description
31	<b>swxhc0s5rstb.</b> Read-write. Reset: 1. <b>Description:</b> 0: XHC0 PHY is powered down. 1: XHC0 PHY is powered.
30	<b>swu2phys5rstb.</b> Read-write. Reset: 1. <b>Description:</b> 0: USB2 PHY is powered down. 1: USB2 PHY is powered. (See NOTE2 for the relationship between USB2 PHY and XHC controller)
29	Reserved.
28	<b>swsatalockphyif.</b> Read-write. Reset: 0. <b>Description:</b> 0: SATA PHY interface is not locked 1: SATA PHY interface is locked
27	<b>swusbpllrstb.</b> Read-write. Reset: 1. <b>Description:</b> SW can program this bit to toggle these USB related reset signals: oUsbPllRstB oUsb3PllRstB oUsbPllLockB oUsbDllRstB oGoodClkForUsb If SwUsbPllRstB is cleared to 0, the above signals will be driven to 0. After SW set SwUsbPllRstB to 1, HW will generate a de-assertion sequence for the above signals.
26	<b>swu3phys5rstb.</b> Read-write. Reset: 1. <b>Description:</b> 0: USB3 PHY is powered down. 1: USB3 PHY is powered. (See NOTE2 for the relationship between USB2 PHY and XHC controller)
25	Reserved.
24	<b>spi_pad_disable.</b> Read-write. Reset: 0. SW set this bit to 1 to disable the OE and PU of the SPI pads.
23	<b>lpc_pad_disable.</b> Read-write. Reset: 0. SW set this bit to 1 to disable the OE and PU of the LPC pads.
22:16	<b>swblockusbrst.</b> Read-write. Reset: 00h. <b>Description:</b> SW can set these bits to 1 to individually block the reset signals to USB2 and USB3. Combined with SwUsbPllRstB bit, SW can control the way it assert reset for USB. Bit 0: Block oUsbPllRstB Bit 1: Block oUsb3PllRstB Bit 2: Block oUsbPllLockB Bit 3: Block oUsbDllRstB Bit 4: Block oGoodClkForUsb Bit 5: Block oSoftKB_2PciRst Bit 6: Block oS3PciRstB
15	<b>powerallpwrland.</b> Read-write. Reset: 1. <b>Description:</b> If PowerAllPwrIsland is 1, we will sequentially power up all the power groups to avoid current surge. If the platform support AOAC, SW has to set this bit to 0 during boot-up. Otherwise, SW can leave it at 1. (See diagram in the NOTE1 below)
14	Reserved.
13	<b>swxhc1s5rstb.</b> Read-write. Reset: 1. <b>Description:</b> 0: XHC1 PHY is powered down. 1: XHC1 PHY is powered.

12	<b>swotgs5rstb.</b> Read-write. Reset: 0. <b>Description:</b> 0: Change OtgS5RstB signal to 0 1: Change OtgS5RstB signal to 1
11:9	Reserved.
8	<b>swsatablockoob_aoac.</b> Read-write. Reset: 0. <b>Description:</b> 0: SATA OOB is not blocked 1: SATA OOB is blocked
7:6	Reserved.
5	<b>gbepwren.</b> Read-write. Reset: 1. <b>Description:</b> Control the power of GBE power island 0: Gbe is powered down 1: Gbe is powered up
4	<b>otgpwren.</b> Read-write. Reset: 1. <b>Description:</b> Control the power of OTG power island 0: Otg is powered down 1: Otg is powered up
3	<b>xlcpwren.</b> Read-write. Reset: 1. <b>Description:</b> Control the power of SSIC power island 0: Ssic is powered down 1: Ssic is powered up
2	<b>imcpwren.</b> Read-write. Reset: 0. <b>Description:</b> Control the power of IMC 0: IMC is powered down 1: IMC is powered up
1	<b>pg2pwren.</b> Read-write. Reset: 1. <b>Description:</b> Control the power of PG2 power island 0: PG2 is powered down 1: PG2 is powered up
0	<b>pg1apwren.</b> Read-write. Reset: 1. <b>Description:</b> Control the power of PG1a power island 0: PG1a is powered down 1: PG1a is powered up

**AOACx000000A4 (FCH::AOAC::zstated3coldmask)**

Read-write. Reset: FBEE\_E61Fh.

\_aliasHOST; AOACx000000A4; AOAC=FED8\_1E00h

Bits	Description
31:0	<b>zstated3coldmask.</b> Read-write. Reset: FBEE_E61Fh. <b>Description:</b> Mask device which don't need to go D3 when System go Low Power State 1: Mask, which means dev no need to go D3 0: No Mask, which means dev need to go D3

**AOACx000000E0 (FCH::AOAC::aoac\_zpr\_cntl\_0)**

Reset: 0000\_0000h.

\_aliasHOST; AOACx000000E0; AOAC=FED8\_1E00h

Bits	Description
31:0	Reserved.

**AOACx000000E4 (FCH::AOAC::aoac\_zpr\_cntl\_1)**

Reset: 0000\_0000h.

\_aliasHOST; AOACx000000E4; AOAC=FED8\_1E00h

**Bits Description**

31:0 Reserved.

**AOACx000000E8 (FCH::AOAC::aoac\_zpr\_cntl\_2)**

Reset: 0000\_0000h.

\_aliasHOST; AOACx000000E8; AOAC=FED8\_1E00h

**Bits Description**

31:0 Reserved.

**AOACx000000EC (FCH::AOAC::aoac\_zpr\_cntl\_3)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; AOACx000000EC; AOAC=FED8\_1E00h

**Bits Description**

31:0 Reserved.

**AOACx000000F0 (FCH::AOAC::shdwreg\_debug)**

\_aliasHOST; AOACx000000F0; AOAC=FED8\_1E00h

Bits	Description
31	Reserved.
30	<b>shadow_hang.</b> Read-only. Reset: 0. <b>Description:</b> We have a timer to keep track of the save/restore process. When save/restore is on-going, the timer will be counting up. When there is no save/restore, the timer is reset to 0. SHADOW_hang bit goes high when the timer timeout and save/restore is not done yet. SW can toggle SHADOW_rst bit to reset the shadow register control logic if the hang condition is detected. 0: Save/restore has been on-going for less than 8~12 ms 1: Save/restore has been on-going for at least 8~12 ms Note: SHADOW_hang means somehow the current request is not completed. SW should do a SHADOW_rst and request again. The Shadow Reg Status bit should remain previous state from last successful save/restore of the corresponding IP. So we don't want to reset Shadow Reg Status bit by SHADOW_rst.
29:0	<b>shadow_err.</b> Read-write. Reset: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXb. <b>Description:</b> Write-1-clear. Write 0 has no effect. This register report errors encountered during each device's save/restore. The following error conditions will be recorded: AHB bus response error PCI bus master abort PCI bus slave abort Bit29: Error condition encountered during device 29 save/restore Bit28: Error condition encountered during device 28 save/restore Bit27: Error condition encountered during device 27 save/restore Bit26: Error condition encountered during device 26 save/restore Bit25: Error condition encountered during device 25 save/restore Bit24: Error condition encountered during device 24 save/restore Bit23: Error condition encountered during device 23 save/restore Bit22: Error condition encountered during device 22 save/restore Bit21: Error condition encountered during device 21 save/restore Bit20: Error condition encountered during device 20 save/restore Bit19: Error condition encountered during device 19 save/restore Bit18: Error condition encountered during device 18 save/restore Bit17: Error condition encountered during device 17 save/restore Bit16: Error condition encountered during device 16 save/restore Bit15: Error condition encountered during device 15 save/restore Bit14: Error condition encountered during device 14 save/restore Bit13: Error condition encountered during device 13 save/restore Bit12: Error condition encountered during device 12 save/restore Bit11: Error condition encountered during device 11 save/restore Bit10: Error condition encountered during device 10 save/restore Bit9 : Error condition encountered during device 9 save/restore Bit8 : Error condition encountered during device 8 save/restore Bit7 : Error condition encountered during device 7 save/restore Bit6 : Error condition encountered during device 6 save/restore Bit5 : Error condition encountered during device 5 save/restore Bit4 : Error condition encountered during device 4 save/restore Bit3 : Error condition encountered during device 3 save/restore Bit2 : Error condition encountered during device 2 save/restore Bit1 : Error condition encountered during device 1 save/restore Bit0 : Error condition encountered during device 0 save/restore

**AOACx000000F4 (FCH::AOAC::pwrrst\_debug)**

Read-only.

NOTE1: When SW read 1 at oPG1a\_PwrGood, oPG2\_PwrGood, oGbe\_PwrGood, oOtg\_PwrGood or oSsic\_PwrGood, that means power-on sequence has completed for power group PG1a, PG2, GBE, OTG or SSIC, respectively.

NOTE2: When SW read 1 at PG1a\_CtrlSts, PG2\_CtrlSts, Gbe\_CtrlSts, Otg\_CtrlSts or Ssic\_CtrlSts, that means power-down sequence has completed for power group PG1a, PG2, GBE, OTG or SSIC, respectively.

\_aliasHOST; AOACx000000F4; AOAC=FED8\_1E00h

Bits	Description
31:0	<p><b>pwrrstdebugbus.</b> Read-only. Reset: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXb.</p> <p><b>Description:</b> Debug signals from PwrRst block:</p> <p>Bit31: Gbe_CtrlSts (see NOTE2)</p> <p>Bit30: PG1a_CtrlSts (see NOTE2)</p> <p>Bit29: PG2_CtrlSts (see NOTE2)</p> <p>Bit28: Ssic_CtrlSts (see NOTE2)</p> <p>Bit27: Otg_CtrlSts (see NOTE2)</p> <p>Bit26: oDebouncedRsmRstB</p> <p>Bit25: oPciRstB</p> <p>Bit24: oCpuPwrGood</p> <p>Bit23: oCpuRstB</p> <p>Bit22: oU2PHY5RstB</p> <p>Bit21: oUsb3S5RstB</p> <p>Bit20: oPllRstB</p> <p>Bit19: oPllLock</p> <p>Bit18: oGbe_PwrGood (see NOTE1)</p> <p>Bit17: oPG1a_PwrGood (see NOTE1)</p> <p>Bit16: oPG2_PwrGood (see NOTE1)</p> <p>Bit15: oSsic_PwrGood (see NOTE1)</p> <p>Bit14: oOtg_PwrGood (see NOTE1)</p> <p>Bit13: oPG1_ToAllEn</p> <p>Bit12: oPG1a_ToAllEn</p> <p>Bit11: oPG2_ToAllEn</p> <p>Bit10: oSsic_ToAllEn</p> <p>Bit9 : oSlpRstAsrtDone</p> <p>Bit8 : S0A3RstDsrtDone</p> <p>Bit7 : S0ResetB_Osc</p> <p>Bit6 : S5ResetB_Osc</p> <p>Bit5 : oDebouncedUsrRstB</p> <p>Bit4 : DevPwrRstB[iDebugSelDev][4]</p> <p>Bit3 : DevPwrRstB[iDebugSelDev][3]</p> <p>Bit2 : DevPwrRstB[iDebugSelDev][2]</p> <p>Bit1 : DevPwrRstB[iDebugSelDev][1]</p> <p>Bit0 : DevPwrRstB[iDebugSelDev][0]</p>



**AOACx000000F8 (FCH::AOAC::sw\_semaphore)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; AOACx000000F8; AOAC=FED8\_1E00h

Bits	Description
31	<b>drvrownaoacreg.</b> Read-write. Reset: 0. <b>Description:</b> Read only. Driver write Set_DrvrOwnAoacReg and Clr_DrvrOwnAoacReg bits to set or clear DrvrOwnAoacReg. When DrvrOwnAoacReg is 1, BiosOwnAoacReg will stay at 0 and cannot be written. Driver should always make sure DrvrOwnAoacReg is 1 before it writes AOAC Registers, and clear DrvrOwnAoacReg after it finishes the programming.
30	Reserved.
29	<b>clr_drvrownaoacreg.</b> Read-write. Reset: 0. Write only. Driver write 1 to this bit to clear DrvrOwnAoacReg.
28	<b>set_drvrownaoacreg.</b> Read-write. Reset: 0. Write only. Driver write 1 to this bit to set DrvrOwnAoacReg.
27	<b>biosownaoacreg.</b> Read-write. Reset: 0. <b>Description:</b> Read only. BIOS write Set_BiosOwnAoacReg and Clr_BiosOwnAoacReg bits to set or clear BiosOwnAoacReg. When BiosOwnAoacReg is 1, DrvrOwnAoacReg will stay at 0 and cannot be written. BIOS should always make sure BiosOwnAoacReg is 1 before it writes AOAC Registers, and clear BiosOwnAoacReg after it finishes the programming.
26	Reserved.
25	<b>clr_biosownaoacreg.</b> Read-write. Reset: 0. Write only. BIOS write 1 to this bit to clear BiosOwnAoacReg.
24	<b>set_biosownaoacreg.</b> Read-write. Reset: 0. Write only. BIOS write 1 to this bit to set BiosOwnAoacReg.
23:0	Reserved.

**AOACx000000FC (FCH::AOAC::aoac\_debug)**

Read-write. Reset: 0080\_0000h.

\_aliasHOST; AOACx000000FC; AOAC=FED8\_1E00h

Bits	Description
31	<b>fch_zsc_wake.</b> Read-write. Reset: 0. Read only
30	<b>fch_zsc_idle.</b> Read-write. Reset: 0. Read only
29	<b>zsc_fch_fence_req.</b> Read-write. Reset: 0. Read only
28	<b>fch_zsc_fence_ack.</b> Read-write. Reset: 0. Read only
27	<b>zsc_fch_cgpll_power_down.</b> Read-write. Reset: 0. Read only
26	<b>fch_zsc_cgpll_lock.</b> Read-write. Reset: 0. Read only
25	<b>shdwm_aclk_enable.</b> Read-write. Reset: 0. Read only
24	<b>shdwm_any_request.</b> Read-write. Reset: 0. Read only
23	<b>shdwm_sram_awake.</b> Read-write. Reset: 1. Read only
22	<b>shdwm_shadowclkon_s.</b> Read-write. Reset: 0. Read only
21	<b>shdwm_shadowinprogress_s.</b> Read-write. Reset: 0. Read only
20	<b>shdwm_ahbm32_aclk_enbf.</b> Read-write. Reset: 0. Read only
19	<b>shdwm_ahbm32_aclk_done.</b> Read-write. Reset: 0. Read only
18	<b>shdwm_ahbm32_aclk_req.</b> Read-write. Reset: 0. Read only
17:13	<b>shdwm_ctrlrid_4_0.</b> Read-write. Reset: 00h. Read only
12:8	<b>shdwm_state_4_0.</b> Read-write. Reset: 00h. Read only
7	<b>s0i3_xtalpciclk_en.</b> Read-write. Reset: 0. Read only
6:4	<b>s0i3_pd_st_2_0.</b> Read-write. Reset: 0h. Read only
3:0	<b>s0i3_state_3_0.</b> Read-write. Reset: 0h. Read only

## **7.3.8 ISA Bridge**

### **7.3.8.1 eSPI Registers**

<b>ESPIx00000000 (FCH::ITF::ESPI::DN_TXHDR_0th)</b>	
Reset: 0020_0000h.	
_aliasHOST; ESPIx00000000; ESPI=FEC2_0000h	
<b>Bits</b>	<b>Description</b>
31:24	<b>DNCMD_HDATA2.</b> Read-write. Reset: 00h. <b>Description:</b> The definition for this field is depended on SW_CMD_TYPE Independent channel command selected: Reserved, Always be 00h Peripheral selected : Length[7:0] VW selected: Reserved, Always be 00h OOB selected: Length[7:0] FLASH selected: Length[7:0]
23:16	<b>DNCMD_HDATA1.</b> Read-write. Reset: 20h. <b>Description:</b> The definition for this field is depended on SW_CMD_TYPE Independent channel command selected: Addres [7:0] of SET_CONFIGURATION/GET_CONFIGURATION Bit[1:0] needs to be 00. Note: In-Band command. These bits are ignored. Peripheral selected [23:20]: Tag [19:16]: Length[11:8] VW selected: Reserved, Always be 00h OOB selected: [23:20]: Tag [19:16]: Length[11:8] FLASH selected: [23:20]: Tag [19:16]: Length[11:8]
15:8	<b>DNCMD_HDATA0.</b> Read-write. Reset: 00h. <b>Description:</b> The definition for this field is depended on SW_CMD_TYPE. Independent command selected: Address [15:8] of SET_CONFIGURATION/GET_CONFIGURATION. [15:12]:0h [11:8]: address[11:8] Note: In-Band command. These bits are ignored. Peripheral selected: a) SW programs this byte to be Message cycle type(0001xxxy) to instruct the eSPI controller send down peripheral message with data(8 bytes+data byte N) or without data(total 8 bytes) b) SW programs this byte to be 'nsuccessful completion to instruct the eSPI controller send down unsuccessful completion VW selected: It indicates the Virtual Wire Count will be send down. Bit[5:0] represends how many Virtual Wire groups to be commnunicated in the same packets. NOTE: In the current design, it is limited to 16 groups(bit[5:4]=00) to save the registers needed. Only 8 indexes are defined In current eSPI spec v0.7. OOB selected: SW programs this byte to be 0x21 to instruct the eSPI controller send send down Tunneled SMBUS message to slave FLASH selected: For MAFS:SW prgrames this byte to be CycleType for Flash Completion, including Cpl/Unsuccessful Cpl/CplD. For SAFS:SW prgrames this byte to be CycleType for FLASH read/write/erase
7:5	Reserved.
4	<b>PUT_FLASH_NP_TRAN_ACTIVE.</b> Read-only. Reset: 0.

	<p><b>Description:</b> SW i/f programming PUT_FLASH_NP transmission active bit. This bit tells whether the PUT_FLASH_NP transmit is done or not. Generally it shall be set once bit [3] is set. When the PUT_FLASH_NP is deferred, bit[3] is cleared but this bit shall keep asserted until the deferred transaction is complete.</p> <p>0b: No PUT_FLASH_NP transmit on-going.</p> <p>1b: PUT_FLASH_NP_transmit on-going.</p>
3	<p><b>DNCMD_STATUS.</b> Read,Write-1-only. Reset: 0.</p> <p><b>Description:</b> Set: The bit needs to be set last by Software after all eSPI specific registers are all programmed to inform the protocol layer to send down command or packet.</p> <p>Clear: Hardware will automatically clear this bit after the packet is sent down.</p> <p>Note: For deferred SAFS downstream command, this bit will clear but 0x00[4] will still be 1 until completion.</p>
2:0	<p><b>DNCMD_TYPE.</b> Read-write. Reset: 0h.</p> <p><b>Description:</b> TX Command Type:</p> <p>000: Set Configuration (Independent command)</p> <p>001: Get Configuration (Independent command)</p> <p>010: In-band RESET command (Independent command)</p> <p>011: reserved</p> <p>100: Peripheral Channel message down stream</p> <p>101: VW Channel down stream</p> <p>110: OOB Channel down stream</p> <p>111: For MAFS:Flash Channel Cpl/CplD/Unsuccessful Cpl down stream. For SAFS:Flash Channel Request downstream</p>

**ESPIx00000004 (FCH::ITF::ESPI::DN\_TXHDR\_1)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000004; ESPI=FEC2\_0000h

Bits	Description
31:24	<b>DNCMD_HDATA6.</b> Read-write. Reset: 00h. <b>Description:</b> The definition for this field is depended on SW_CMD_TYPE Independent channel command selected: data[31:24] Peripheral selected : Message specific byte 2 VW selected: Reserved, Always be 00h OOB selected: Reserved, Always be 00h FLASH selected: For MAFS: Reserved, Always be 00h For SAFS: If Cycle Type is FLASH_WRITE, FLASH_READ, FLAS_ERASE, this field is Address[31:24]. If Cycle Type is FLASH_RPMC_OP1 (R1R0 = 0x0) or FLASH_RPMC_RPMC_OP2, this field is reserved. If Cycle Type is FLASH_RPMC_OP1 (R1R0 = 0x1/2/3), this field is data byte 0.
23:16	<b>DNCMD_HDATA5.</b> Read-write. Reset: 00h. <b>Description:</b> The definition for this field is depended on SW_CMD_TYPE Independent channel command selected: data[23:16] Peripheral selected : Message specific byte 1 VW selected: Reserved, Always be 00h OOB selected: SMBus Byte Count. Need to program not greater than 128bytes FLASH selected: For MAFS: Reserved, Always be 00h For SAFS: If Cycle Type is FLASH_WRITE, FLASH_READ, FLAS_ERASE, this field is Address[23:16]. If Cycle Type is FLASH_RPMC_OP1 (R1R0 = 0x0) or FLASH_RPMC_RPMC_OP2, this field is reserved. If Cycle Type is FLASH_RPMC_OP1 (R1R0 = 0x1/2/3), this field is data byte 1.
15:8	<b>DNCMD_HDATA4.</b> Read-write. Reset: 00h. <b>Description:</b> The definition for this field is depended on SW_CMD_TYPE Independent channel command selected: data[15:8] Peripheral selected : Message specific byte 0 VW selected: Reserved, Always be 00h OOB selected: SMBus Command Op Code. FLASH selected: For MAFS: Reserved, Always be 00h For SAFS: If Cycle Type is FLASH_WRITE, FLASH_READ, FLAS_ERASE, this field is Address[15:8]. If Cycle Type is FLASH_RPMC_OP1 (R1R0 = 0x0) or FLASH_RPMC_RPMC_OP2, this field is reserved. If Cycle Type is FLASH_RPMC_OP1 (R1R0 = 0x1/2/3), this field is data byte 2.
7:0	<b>DNCMD_HDATA3.</b> Read-write. Reset: 00h.

	<p><b>Description:</b> The definition for this field is depended on SW_CMD_TYPE</p> <p>Independent channel command selected: data[7:0]</p> <p>Peripheral selected : Message code [7:0]</p> <p>VW selected: Reserved, Always be 00h</p> <p>OOB selected:</p> <p>SMBus Slave Address. Bit[0] needs to program to 1</p> <p>FLASH selected:</p> <p>For MAFS:</p> <p>Reserved, Always be 00h</p> <p>For SAFS:</p> <p>If Cycle Type is FLASH_WRITE, FLASH_READ, FLAS_ERASE, this field is Address[7:0].</p> <p>If Cycle Type is FLASH_RPMC_OP1 (R1R0 = 0x0) or FLASH_RPMC_RPMC_OP2, this field is reserved.</p> <p>If Cycle Type is FLASH_RPMC_OP1 (R1R0 = 0x1/2/3), this field is data byte 3.</p>
--	---

#### ESPIx00000008 (FCH::ITF::ESPI::DN\_TXHDR\_2)

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000008; ESPI=FEC2\_0000h

Bits	Description
31:8	Reserved.
7:0	<p><b>DNCMD_HDATA7.</b> Read-write. Reset: 00h.</p> <p><b>Description:</b> The definition for this field is depended on SW_CMD_TYPE</p> <p>Independent channel command selected: Reserved, Always be 00h</p> <p>Peripheral selected : Message specific byte 3</p> <p>VW selected: Reserved, Always be 00h</p> <p>OOB selected: Reserved, Always be 00h</p> <p>FLASH selected: Reserved, Always be 00h</p>

**ESPIx0000000C (FCH::ITF::ESPI::DN\_TXDATA\_PORT)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000000C; ESPI=FEC2\_0000h

Bits	Description
31:24	<b>DN_TXDATA_B3.</b> Read-write. Reset: 00h. <b>Description:</b> The definition for this field is depended on SW_CMD_TYPE Independent channel command selected: Reserved, Always be 00h Peripheral selected : Message Data DWn[31:24] VW selected: VW Index Group 2n+1 data OOB selected: OOB Message DWn[31:24] FLASH selected: For MAFS:Flash Cpl Data DWn[31:24] For SAFS:Flash write data [31:24]
23:16	<b>DN_TXDATA_B2.</b> Read-write. Reset: 00h. <b>Description:</b> The definition for this field is depended on SW_CMD_TYPE Independent channel command selected: Reserved, Always be 00h Peripheral selected : Message Data DWn[23:16] VW selected: VW Index Group 2n+1 OOB selected: OOB Message DWn[23:16] FLASH selected: For MAFS:Flash Cpl Data DWn[23:16] For SAFS:Flash write data [23:16]
15:8	<b>DN_TXDATA_B1.</b> Read-write. Reset: 00h. <b>Description:</b> The definition for this field is depended on SW_CMD_TYPE Independent channel command selected: Reserved, Always be 00h Peripheral selected : Message Data DWn[15:8] VW selected: VW Index Group 2n Data OOB selected: OOB Message DWn[15:8] FLASH selected: For MAFS:Flash Cpl Data DWn[15:8] For SAFS:Flash write data [15:8]
7:0	<b>DN_TXDATA_B0.</b> Read-write. Reset: 00h. <b>Description:</b> The definition for this field is depended on SW_CMD_TYPE Independent channel command selected: Reserved, Always be 00h Peripheral selected : Message Data DWn[7:0] VW selected: VW Index Group 2n OOB selected: OOB Message DWn[7:0] FLASH selected: For MAFS:Flash Cpl Data DWn[7:0] For SAFS:Flash write data [7:0]

**ESPIx00000010 (FCH::ITF::ESPI::UP\_RXHDR\_0)**

Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000010; ESPI=FEC2\_0000h

Bits	Description
31:24	<b>UPCMD_HDATA2.</b> Read-only. Reset: 00h. <b>Description:</b> RX_LOW_LEN: This field store the Length[7:0] from GET_FLASH_NP(for MAFS)/GET_OOB/GET_FLASH_C(for SAFS) [31:24] Length[7:0]
23:16	<b>UPCMD_HDATA1.</b> Read-only. Reset: 00h. <b>Description:</b> RX_TAG_LEN: This field store the Tag and Length[11:8] which from eSPI packet received by GET_FLASH_NP(for MAFS)/GET_OOB/GET_FLASH_C(for SAFS) [23:20], Tag [19:16] Length[11:8]
15:8	<b>UPCMD_HDATA0.</b> Read-only. Reset: 00h. <b>Description:</b> Cycle Type This field store the cycle type from GET_FLASH_NP(for MAFS) and GET_OOB This field store the cycle type from GET_FLASH_C(for SAFS) and PUT_FLASH_NP with immediately response completion(for SAFS)
7:6	Reserved.
5:4	<b>SLAVE_SEL.</b> Read-only. Reset: 0h. <b>Description:</b> Slave N Received selected 00: The upstream packet is from Slave0 Others: Reserved
3	<b>UPCMD_STATUS.</b> Read,Write-1-to-clear. Reset: 0. <b>Description:</b> For OOB channel, this bit will be set after received OOB message packet. For Flash channel, there are MAFS mode and SAFS mode: MAFS: this bit will be set after received Flash request packet. SAFS: this bit will be set after received Flash completion only after SW programming interface sending PUT_FLASH_NP. 1. When SW programming interface sending PUT_FLASH_NP cycle with immediately response completion or 2. Received completion by GET_FLASH_C for SW programming interface sending PUT_FLASH_NP. (For SAFS channel, GET_FLASH_C will not assert this bit) Note:eSPI will not send down another GET_OOB or GET_FLASH_NP(MAFS) or GET_FLASH_C(For SW programming interface sending PUT_FLASH_NP) before the Valid bit cleared by SW. This bit can be cleared by SW writing 1'b1 to this field.
2:0	<b>UPCMD_TYPE.</b> Read-only. Reset: 0h. <b>Description:</b> 000: For MAFS, Flash Channel Request (GET_FLASH_NP) 001: Upstream OOB message (GET_OOB) 010: For SAFS, Flash Channel completion (GET_FLASH_C) 011: For SAFS, Current PUT_FLASH_NP Cycle response completion with header/data. Note: This value is only valid when PUT_FLASH_NP_HEADER_DATA_EN =1 or PUT_FLASH_NP_HEADER_EN =1 100: For SAFS, GET_STATUS with modifier. Others: Reserved.



**ESPIx00000014 (FCH::ITF::ESPI::UP\_RXHDR\_1)**

Read-only. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000014; ESPI=FEC2\_0000h

Bits	Description
31:24	<b>UPCMD_HDATA6.</b> Read-only. Reset: 00h. <b>Description:</b> The definition for this field is depended on RX_REQ_TYPE OOB selected: Reserved FLASH selected(for MAFS only): Address[7:0]
23:16	<b>UPCMD_HDATA5.</b> Read-only. Reset: 00h. <b>Description:</b> The definition for this field is depended on RX_REQ_TYPE OOB selected: SMBus Byte Count FLASH selected(for MAFS only): Address[15:8]
15:8	<b>UPCMD_HDATA4.</b> Read-only. Reset: 00h. <b>Description:</b> The definition for this field is depended on RX_REQ_TYPE OOB selected: SMBus Command Opcode FLASH selected(for MAFS only): Address[23:16]
7:0	<b>UPCMD_HDATA3.</b> Read-only. Reset: 00h. <b>Description:</b> The definition for this field is depended on RX_REQ_TYPE OOB selected SMBus Slave Address FLASH selected(for MAFS only): Address[31:24]

**ESPIx00000018 (FCH::ITF::ESPI::UP\_RXDATA\_PORT)**

Read-only. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000018; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>UP_RXDATA.</b> Read-only. Reset: 0000_0000h.

**ESPIx0000002C (FCH::ITF::ESPI::MASTER\_CAP)**

Read-only. Reset: Fixed,E849\_EDAFh.

\_aliasHOST; ESPIx0000002C; ESPI=FEC2\_0000h

Bits	Description
31	<b>CRC_CHECK_SUPPORT.</b> Read-only. Reset: Fixed,1.
30	<b>ALERT_MODE_SUPPORT.</b> Read-only. Reset: Fixed,1. A dedicated Alert# pin is used to signal the Alert event, or I/O[1] pin used for Alert.
29:28	<b>IO_MODE_SUPPORT.</b> Read-only. Reset: Fixed,2h. IO Mode support by Controller, Quad mode, Dual mode, single mode
27:25	<b>CLK_FREQ_SUPPORT.</b> Read-only. Reset: Fixed,4h. <b>Description:</b> Operating Support Frequency 000 20MHz 001 16.7MHz, 33MHz 011 16.7MHz, 33MHz, 66MHz 100 16.7MHz, 20MHz, 25MHz, 33MHz, 50MHz, 66MHz Note: This field has a default value of 000 to reflect tINIT-FREQ (Table21) of 20MHz max.
24:22	<b>SLAVE_NUM.</b> Read-only. Reset: Fixed,1h.
21:19	<b>PR_MAX_SIZE.</b> Read-only. Reset: Fixed,1h. 64 bytes address aligned max payload size.
18:13	<b>VW_MAX_SIZE.</b> Read-only. Reset: Fixed,0Fh. Operating Maximum Virtual Wire Count support
12:10	<b>OOB_MAX_SIZE.</b> Read-only. Reset: Fixed,3h. 256 bytes max payload size
9:7	<b>FLASH_MAX_SIZE.</b> Read-only. Reset: Fixed,3h. 256 bytes max payload size
6:4	<b>ESPI_VERSION.</b> Read-only. Reset: Fixed,2h. ESPI Version. 000b:Master support eSPI 0.7 version; 001b:Master support eSPI 0.75 version; 010b:Master support eSPI 1.0 version;
3	<b>PR_SUPPORT.</b> Read-only. Reset: Fixed,1.
2	<b>VW_SUPPORT.</b> Read-only. Reset: Fixed,1.
1	<b>OOB_SUPPORT.</b> Read-only. Reset: Fixed,1.
0	<b>FLASH_SUPPORT.</b> Read-only. Reset: Fixed,1.

**ESPIx00000030 (FCH::ITF::ESPI::GLOBAL\_CONTROL\_0)**

Read-write. Reset: 0000\_0008h.

\_aliasHOST; ESPIx00000030; ESPI=FEC2\_0000h

Bits	Description
31	<b>SAFS_Clk_Gate_EN.</b> Read-write. Reset: 0. When this bit is set, enable dynamic clock gating for boot from SAFS channel. And SAFS through register i/f is not impacted by this bit. Default is disable.
30	<b>PR_RST_EN_PLTRST.</b> Read-write. Reset: 0. This bit is used to control whether to reset ESPI peripheral channel when ACPI_PLTRSTB is received. Default value is 0 and not to reset
29:24	<b>WAIT_CNT.</b> Read-write. Reset: 00h. Specifies the timeout count for wait state. This value means the threshold of wait state timeout. For example, if program theses bits to 6'h11 (6'd17), it means timeout will be fired if device inserts 17 byte wait time, then host will disable eSPI_Clk output. By default, host will allow at most 16 byte wait time, and will time out if device inserts 17 bytes wait time, then eSPI_Clk will be disabled.
23:8	<b>WDG_CNT.</b> Read-write. Reset: 0000h. Specifies the timeout retry count for PCI downstream retries.
7	<b>RG_dbgclk_gating_en.</b> Read-write. Reset: 0.
6:4	<b>AL_IDLE_TIMER.</b> Read-write. Reset: 0h. <b>Description:</b> Set the bits to select different Idle timer timeout value. Once the Idle timer reach the timeout value and Global Alink clock gating Enable set, eSPI will output ESPI_Stop_AIClk to do global Alink clock gating. Bits Selection 000 16 clocks 001 32 clocks 010 64 clocks 011 128 clocks 100 256 clocks 101 512 clocks 110 1024 clocks 111 2048 clocks
3	<b>AL_STOP_EN.</b> Read-write. Reset: 1. Set the bit to enable eSPI generating ESPI_Stop_AIClk to do global Alink clock gating once Global Alink Idle Timer reach the timeout value.
2	<b>PR_CLKGAT_EN.</b> Read-write. Reset: 0. Set the bit to enable Peripheral block do dynamic clock gating once the Slave Peripheral channel is disabled.
1	<b>WAIT_CHKEN.</b> Read-write. Reset: 0. Set the bit to enable the Wait State counter during eSPI bus turn around.
0	<b>WDG_EN.</b> Read-write. Reset: 0. Set the bit to enable the watchdog counter for all the PCI downstream transactions for eSPI.

**ESPIx00000034 (FCH::ITF::ESPI::GLOBAL\_CONTROL\_1)**

Read-write.

\_aliasHOST; ESPIx00000034; ESPI=FEC2\_0000h

Bits	Description
31:22	Reserved.
21	<b>ESPI_REQ_NOTWITH_VW_REQ.</b> Read-write. When 0: It applies to the case where eSPI host controller only sits onto one data bus with device. Then if eSPI_VW_Req gets granted, no eSPI_Req shall need to be asserted to the same SPI controller further. When set to 1: It applies to the case where eSPI host controller might sit onto two data bus with device based on one SPI register setting. eSPI_Req and eSPI_VW_Req don't go to the same data bus. Then even if eSPI_VW_Req gets granted, eSPI_Req is still needed to be asserted to the related SPI controller further before issuing any transaction (like PUT_VM).
20	<b>ALERT_ENABLE.</b> Read-write. Reset: 0. When 0b: ALERT# to eSPI keep inactive(1). When set to 1b: Alert# value is monitored from Alert# or Din[1] depending on Alert_mode setting.
19	<b>FL_REQ_VALID_DIS.</b> Read-write. Reset: 0. When 0b: 1 bit[5:4] of VW index45 are treated as valid bits for FL_REQ/FL_REQ_ATOMIC. 2. bit[4] of VW index 43h, the valid bit will be 1 when sending down FL_GNT.
18	<b>ROM_SHARING_DIS.</b> Read-write. Reset: 0. When 0b, it means: 1. GET_VW will be sent down to eSPI bus as long as VW available status in GET_STATUS is set to 1 no matter VW channel is enabled or not. 2. If the GET_VW without FL_Req bit set, this GET_VW should be discarded. 3. When receiving eSPI_VW_GNT indicating that SPI has granted this FL_Req and already tri-stated the bus, eSPI controller will unconditionally send down PUT_VW with the FL_ACK set without requesting SPI the bus usage (i.e. no eSPI_Req is asserted to SPI controller if bit [21] is 0b, otherwise, eSPI_Req is required to assert before issuing PUT_VW). When 1b, it means to disable ROM access: 1. Get_VW/Put_VW will follow the normal eSPI protocol, i.e., the VW channel need be enabled first. 2. No eSPI_VW_Req/eSPI_VW_Atomic will be sent to eSPI host controller. 3. No response to eSPI_VW_GNT from eSPI controller.
17:13	<b>RGCMD_INT_MAP.</b> Read-write. Reset: 17h. <b>Description:</b> Register CMD interrupt mapping. When Register command (Downstream/Upstream peripheral message, Downstream/Upstream OOB, Downstream VW, Channel Independent command) have finished, the interrupt eSPI controller generates will be mapped to interrupt pin according to this register setting. Bits Mapping 00000: IRQ0 00001: IRQ1 .... 10111: IRQ23 11111: SMI#
12:8	<b>ERR_INT_MAP.</b> Read-write. Reset: 1Fh. <b>Description:</b> ERR interrupt mapping. When some errors happen to slave transactions and the related error interrupt enable has been set, the error interrupt will be mapped to interrupt pin according to following register setting. Bits Mapping 00000: IRQ0 00001: IRQ1 .... 10111: IRQ23 11111: SMI#
7:3	Reserved.
2	<b>SUB_DECODE_EN.</b> Read-write. Reset: 0. Enable eSPI to do Subtractive Decode.
1	<b>BUS_MASTER_EN.</b> Read-write. Reset: 0. Enable eSPI Upstream Memory cycle posting.
0	<b>SW_RST.</b> Read-write. Reset: 0. Set the bit to do global controller resets for eSPI controller. All the state machines will return to idle, and all the request will be flushed. All the configuration registers will be reset to default value, and software needs to send In-Band Reset to each Slave device after controller reset so that both Master and Slave run in same configuration mode.

**ESPIx00000038 (FCH::ITF::ESPI::SEMAPHORE\_MISC\_CONTROL\_REG0)**

Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000038; ESPI=FEC2\_0000h

Bits	Description
31	<b>espi_p1500_timeout_error_bit.</b> Read-only. Reset: 0. A 1b of this bit means the TDR access to eSPI registers times out. And this bit is real-time. When this bit is 1, it means the last JTAG W/R operation is timeout. This bit will be clear if this time JTAG W/R operation succeed.
30	<b>SW3_OWNER_CLR.</b> Read-write. Reset: 0. Clear SW3 ownership bit, Clear has higher priority than Set. Once eSPI is not needed, SW3 need write 1b to this bit to release the ownership. And this bit need be written back to 0b in the same cycle that bit [29] is set to 1, or at the end of SW0 releasing its ownership.
29	<b>SW3_OWNER_SET.</b> Read-write. Reset: 0. Set SW3 ownership bit. This bit is allowed to write only when all the status bit [15:8], [16], [20], [24], [28] are all 0b. Note: this bit should not be set simultaneously with <Set SW0/SW1/SW2 ownership bit> and SW4_USER_ID.
28	<b>SW3_OWNER_STATUS.</b> Read-only. Reset: 0. SW3 ownership status bit (recommend to use if SW0, SW1, SW2 are also occupied). Note: CLR has higher priority than SET
27	Reserved.
26	<b>SW2_OWNER_CLR.</b> Read-write. Reset: 0. Clear SW2 ownership bit, Clear has higher priority than Set. Once eSPI is not needed, SW2 need write 1b to this bit to release the ownership. And this bit need be written back to 0b in the same cycle that bit [25] is set to 1, or at the end of SW2 releasing its ownership.
25	<b>SW2_OWNER_SET.</b> Read-write. Reset: 0. Set SW2 ownership bit. This bit is allowed to write only when all the status bit [15:8], [16], [20], [24], [28] are all 0b. Note: this bit should not be set simultaneously with <Set SW0/SW1/SW3 ownership bit> and SW4_USER_ID.
24	<b>SW2_OWNER_STATUS.</b> Read-only. Reset: 0. SW2 (recommend X86 to use) ownership status bit. Note: CLR has higher priority than SET
23	Reserved.
22	<b>SW1_OWNER_CLR.</b> Read-write. Reset: 0. Clear SW1 ownership bit, Clear has higher priority than Set. Once eSPI is not needed, SW1 need write 1b to this bit to release the ownership. And this bit need be written back to 0b in the same cycle that bit [21] is set to 1, or at the end of SW1 releasing its ownership.
21	<b>SW1_OWNER_SET.</b> Read-write. Reset: 0. Set SW1 ownership bit. This bit is allowed to write only when all the status bit [15:8], [16], [20], [24], [28] are all 0b. Note: this bit should not be set simultaneously with <Set SW0/SW2/SW3 ownership bit> and SW4_USER_ID.
20	<b>SW1_OWNER_STATUS.</b> Read-only. Reset: 0. SW1 (recommend MP1 (SMU) to use) ownership status bit. Note: CLR has higher priority than SET
19	Reserved.
18	<b>SW0_OWNER_CLR.</b> Read-write. Reset: 0. Clear SW0 ownership bit, Clear has higher priority than Set. Once eSPI is not needed, SW0 need write 1b to this bit to release the ownership. And this bit need be written back to 0b in the same cycle that bit [17] is set to 1, or at the end of SW0 releasing its ownership..
17	<b>SW0_OWNER_SET.</b> Read-write. Reset: 0. Set SW0 ownership bit. This bit is allowed to write only when all the status bit [15:8], [16], [20], [24], [28] are all 0b. Note: this bit should not be set simultaneously with <Set SW1/SW2/SW3 ownership bit> and SW4_USER_ID.
16	<b>SW0_OWNER_STATUS.</b> Read-only. Reset: 0. SW0 (recommend MP0 (PSP) to use) ownership status bit. Note: CLR has higher priority than SET

15:8	<b>SW4_USER_ID</b> . Read-write. Reset: 00h. This is a second mechanism to support more players to operate eSPI in a system over 4 players are there. Anyone which need operate eSPI need poll this field to be 0h before writing its own ID to this field, and it can start to operate eSPI once this field is read to be its own ID. And once it doesn't need to operate eSPI, it need write this field back to 0b to release the usage of eSPI. In this mechanism, anyone which need operate eSPI is required to poll this field and all other status bits [16], [20], [24], [28] to be 0h before it is allowed to write its own ID to this field to claim the ownership, and once this field is read to be its own ID, it can start to operate eSPI. And once it doesn't need to operate eSPI, it can write this field back to 0b to release its ownership for eSPI. Users are not allowed to write 0h to this filed unless it owns eSPI, and users should use byte enable to avoid this field is wrongly overridden by 0b if they don't intend to use this filed. On the system level, each player should be assigned a unique non-zero ID. And writing ID to this field won't succeed if this filed is not zero.
7:0	Reserved.

**ESPIx00000044 (FCH::ITF::ESPI::SLAVE0\_IO\_BASE\_REG0)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000044; ESPI=FEC2\_0000h

Bits	Description
31:16	<b>RANGE1</b> . Read-write. Reset: 0000h. IO decode base address for Range 1
15:0	<b>RANGE0</b> . Read-write. Reset: 0000h. IO decode base address for Range 0

**ESPIx00000048 (FCH::ITF::ESPI::SLAVE0\_IO\_BASE\_REG1)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000048; ESPI=FEC2\_0000h

Bits	Description
31:16	<b>RANGE3</b> . Read-write. Reset: 0000h. IO decode base address for Range 3
15:0	<b>RANGE2</b> . Read-write. Reset: 0000h. IO decode base address for Range 2

**ESPIx0000004C (FCH::ITF::ESPI::SLAVE0\_IO\_SIZE)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx0000004C; ESPI=FEC2\_0000h

Bits	Description
31:24	<b>RANGE3</b> . Read-write. Reset: 00h. Programmable IO Range3 size
23:16	<b>RANGE2</b> . Read-write. Reset: 00h. Programmable IO Range2 size
15:8	<b>RANGE1</b> . Read-write. Reset: 00h. Programmable IO Range1 size
7:0	<b>RANGE0</b> . Read-write. Reset: 00h. Programmable IO Range0 size

**ESPIx00000050 (FCH::ITF::ESPI::SLAVE0\_MMIO\_BASE\_REG0)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000050; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>RANGE0</b> . Read-write. Reset: 0000_0000h. MMIO decode base address for Range 0

**ESPIx00000054 (FCH::ITF::ESPI::SLAVE0\_MMIO\_BASE\_REG1)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000054; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>RANGE1</b> . Read-write. Reset: 0000_0000h. MMIO decode base address for Range 1

**ESPIx00000058 (FCH::ITF::ESPI::SLAVE0\_MMIO\_BASE\_REG2)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000058; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>RANGE2</b> . Read-write. Reset: 0000_0000h. MMIO decode base address for Range 2

**ESPIx0000005C (FCH::ITF::ESPI::SLAVE0\_MMIO\_BASE\_REG3)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx0000005C; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>RANGE3.</b> Read-write. Reset: 0000_0000h. MMIO decode base address for Range 3

**ESPIx00000060 (FCH::ITF::ESPI::SLAVE0\_MMIO\_SIZE\_REG0)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000060; ESPI=FEC2\_0000h

Bits	Description
31:16	<b>RANGE1.</b> Read-write. Reset: 0000h. Programmable MMIO Range1 size.
15:0	<b>RANGE0.</b> Read-write. Reset: 0000h. Programmable MMIO Range0 size.

**ESPIx00000064 (FCH::ITF::ESPI::SLAVE0\_MMIO\_SIZE\_REG1)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000064; ESPI=FEC2\_0000h

Bits	Description
31:16	<b>RANGE3.</b> Read-write. Reset: 0000h. Programmable MMIO Range3 size.
15:0	<b>RANGE2.</b> Read-write. Reset: 0000h. Programmable MMIO Range2 size.

**ESPIx00000068 (FCH::ITF::ESPI::SLAVE0\_CONFIG)**

Read-write. Reset: 0000\_0728h.

\_aliasHOST; ESPIx00000068; ESPI=FEC2\_0000h

Bits	Description
31	<b>CRC_CHECK_EN.</b> Read-write. Reset: 0. CRC Checking Enable: This bit is set to 1 by eSPI master to enable the CRC checking on the eSPI bus. 0b: CRC checking is disabled. 1b: CRC checking is enabled.
30	<b>ALERT_MODE_SEL.</b> Read-write. Reset: 0. <b>Description:</b> 0b: IO bit1 pin is used to signal the Alert event. 1b: A dedicated Alert# pin is used to signal the Alert event. Note: This bit can only be 0 in a single master-single slave topology. For single master-multiple slave topology, this bit must be programmed to 1.
29:28	<b>IO_MODE_SEL.</b> Read-write. Reset: 0h. I/O Mode Select. 00 Single I/O 01 Dual I/O 10 Quad I/O 11 Reserved.
27:25	<b>CLK_FREQ_SEL.</b> Read-write. Reset: 0h. <b>Description:</b> Operating Frequency: 000 20MHz; 001 25MHz; 010 33MHz; 011 50MHz; 100 66MHz; 110 16MHz; Others Reserved.
24:12	Reserved.
11	<b>FLASH_modifier_EN.</b> Read-write. Reset: 0. When this bit is set to 1, SAFS FLASH modifier is enable. And if UPCMD_STATUS(0x10[3]) is not cleared, GET_STATUS will not be sent to ESPI bus.
10	<b>SAFS_DEFER_VALID_EN.</b> Read-write. Reset: 1. When this bit is set, that means after PUT_FALSH_NP (flash read/write/erase) is sent to device, Slave response DEFER is a valid behavior and ESPI controller should accept it. When this bit is not set, that means after PUT_FALSH_NP (flash read/write/erase) is sent to device, Slave response DEFER is an invalid behavior and ESPI controller will report invalid response error. This bit default is 1b.
9	<b>PUT_FLASH_NP_HEADER_EN.</b> Read-write. Reset: 1. When this bit is set, that means after PUT_FALSH_NP (flash write/erase) is sent to device, Slave should respond with header (cycle type, tag, length) following Accept byte, which means this request is fulfilled. When this bit is not set, that means after PUT_FALSH_NP (flash read) is sent to device, Slave will respond without header following Accept byte, and controller need still issue additional GET_FALSH_C down to get the completion once it is ready. This bit default is 1b.
8	<b>PUT_FLASH_NP_HEADER_DATA_EN.</b> Read-write. Reset: 1. When this bit is set, that means after PUT_FALSH_NP (flash read) is sent to device, Slave should respond with header (cycle type, tag, length) and data (optional) following Accept byte, which means this request is fulfilled. When this bit is not set, that means after PUT_FALSH_NP (flash read) is sent to device, Slave will respond without header and data following Accept byte, and controller need still issue additional GET_FALSH_C down to get the completion once it is ready. This bit default is 1b.
7:5	<b>FLASH_MPS.</b> Read-write. Reset: 1h. <b>Description:</b> Flash Access Channel Maximum Payload Size Selected, 3 bits: default 001b 000b: Reserved. 001b: 64 bytes max payload size. 010b: 128 bytes max payload size. 011b: 256 bytes max payload size. 100b-111b: Reserved.
4	<b>FLASH_SHARING_MODE.</b> Read-write. Reset: 0. When Flash Access channel is enabled, this bit indicates the flash sharing scheme in operation. 0b: Master attached flash sharing. 1b: Slave attached flash sharing. Default will be 0b
3	<b>PR_EN.</b> Read-write. Reset: 1. Peripheral Channel Enable: This bit is set to 1 by eSPI master to enable the Peripheral channel.
2	<b>VW_EN.</b> Read-write. Reset: 0. Virtual Wire Channel Enable: This bit is set to 1 by eSPI master to enable the Virtual Wire channel.
1	<b>OOB_EN.</b> Read-write. Reset: 0. OOB Message Channel Enable: This bit is set to 1 by eSPI master to enable the OOB Message channel.



0	<b>FLASH_EN.</b> Read-write. Reset: 0. Flash Access Channel Enable: This bit is set to 1 by eSPI master to enable the Flash Access channel.
---	---

**ESPIx0000006C (FCH::ITF::ESPI::SLAVE0\_INT\_EN)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx0000006C; ESPI=FEC2\_0000h

Bits	Description
31	<b>FLASH_REQ_INT_EN.</b> Read-write. Reset: 0. Flash Request Received Enable for X86: Enable to generate a command interrupt to ACPI for X86 when Upstream Flash Request is received and valid to read
30	<b>RXOOB_INT_EN.</b> Read-write. Reset: 0. OOB Message Received Enable: Enable to generate a command interrupt when Upstream OOB Message is received and valid to read
29	<b>RXMSG_INT_EN.</b> Read-write. Reset: 0. Peripheral Message Received Enable: Enable to generate a command interrupt when Upstream Peripheral Msg is received and valid to read.
28	<b>DNCMD_INT_EN.</b> Read-write. Reset: 0. Downstream Register Command Complete Enable: Enable to generate a command interrupt when Downstream eSPI Registers programming Command has completed.
27	<b>RXVW_GRP3_INT_EN.</b> Read-write. Reset: 0. Virtual Wire Index Group 3 Received Enable: Enable to generate a command interrupt when Virtual Wire Index Group3 register specified Virtual Wire Packet is received
26	<b>RXVW_GRP2_INT_EN.</b> Read-write. Reset: 0. Virtual Wire Index Group 2 Received Enable: Enable to generate a command interrupt when Virtual Wire Index Group2 register specified Virtual Wire Packet is received
25	<b>RXVW_GRP1_INT_EN.</b> Read-write. Reset: 0. Virtual Wire Index Group 1 Received Enable: Enable to generate a command interrupt when Virtual Wire Index Group1 register specified Virtual Wire Packet is received
24	<b>RXVW_GRP0_INT_EN.</b> Read-write. Reset: 0. Virtual Wire Index Group 0 Received Enable: Enable to generate a command interrupt when Virtual Wire Index Group0 register specified Virtual Wire Packet is received
23	<b>FLASH_REQ_INT_2_RSMU_EN.</b> Read-write. Reset: 0. Enable to generate a command interrupt to rSMU when Upstream Flash Request (recorded in 0x70[31]), default is disable it.
22:20	Reserved.
19	<b>WDG_TIMEOUT_INT_EN.</b> Read-write. Reset: 0. Alink Bus Watch Dog Timer Timeout Enable: Enable generate an ERROR interrupt when Alink bus Watch dog timer timeout
18	<b>MST_ABORT_INT_EN.</b> Read-write. Reset: 0. Alink Bus Master Abort Enable: Enable generate an ERROR interrupt when eSPI Controller doing Master abort.
17	<b>UPFIFO_WDGTIM_INT_EN.</b> Read-write. Reset: 0. Enable to generate an interrupt when UPFIFO watchdog timer times out.
16	Reserved.
15	<b>PROTOCOL_ERR_INT_EN.</b> Read-write. Reset: 0. Protocol ERROR detected Enable: Enable to generate an ERROR interrupt when Protocol ERROR is detected
14	<b>RXFLASH_OVERFLOW_INT_EN.</b> Read-write. Reset: 0. FLASH Packet Data Length Over 256 bytes Enable: Enable to generate an ERROR interrupt when FLASH Packet data Over 256 bytes
13	<b>RXMSG_OVERFLOW_INT_EN.</b> Read-write. Reset: 0. Peripheral Message Data Length Over 32 bytes Enable: Enable to generate an ERROR interrupt when Peripheral Packet Message data Over 32 bytes
12	<b>RXOOB_OVERFLOW_INT_EN.</b> Read-write. Reset: 0. OOB Packet Data Length Over 256 bytes Enable: Enable to generate an ERROR interrupt when OOB Packet data Over 256 bytes
11	<b>ILLEGAL_LEN_INT_EN.</b> Read-write. Reset: 0. Illegal Response Length Received Enable: Enable generate an ERROR interrupt when Illegal length is received
10	<b>ILLEGAL_TAG_INT_EN.</b> Read-write. Reset: 0. Illegal Response Tag Received Enable: Enable to generate an ERROR interrupt when Illegal tag is received
9	<b>UNSUCSS_CPL_INT_EN.</b> Read-write. Reset: 0. Unsuccessful CPL Received Enable: Enable to generate an ERROR interrupt when Unsuccessful completion without data is received
8	<b>INVALID_CT_INT_EN.</b> Read-write. Reset: 0. Invalid Cycle Type Received Enable: Enable to generate an ERROR interrupt when Unrecognized Cycle Type is received
7	<b>INVALID_RSP_INT_EN.</b> Read-write. Reset: 0. Invalid Response Code Received Enable: Enable to generate an ERROR interrupt when Unrecognized Response is received
6	<b>NON_FATAL_ERR_INT_EN.</b> Read-write. Reset: 0. NON_FATAL_ERROR Response Code Received Enable: Enable to generate an ERROR interrupt when NON_FATAL_ERROR Response Code is received.

5	<b>FATAL_ERR_INT_EN</b> . Read-write. Reset: 0. FATAL_ERROR Response Code Received Enable: Enable to generate an ERROR interrupt when FATAL_ERROR Response Code is received.
4	<b>NO_RSP_INT_EN</b> . Read-write. Reset: 0. NO_RESPONSE Code received Enable: Enable to generate an ERROR interrupt when No_Response Response Code is received.
3	Reserved.
2	<b>CRC_ERR_INT_EN</b> . Read-write. Reset: 0. CRC Error detected Enable: Enable to generate an ERROR interrupt when CRC ERROR is detected on response phase.
1	<b>WAIT_TIMEOUT_INT_EN</b> . Read-write. Reset: 0. eSPI Bus Wait State Insertion Max Out Enable: Enable to generate an ERROR interrupt when eSPI Wait State timer timeout.
0	<b>BUS_ERR_INT_EN</b> . Read-write. Reset: 0. eSPI Bus Timing Error Enable: Enable generate an ERROR interrupt when eSPI Bus timing Error.

**ESPIx00000070 (FCH::ITF::ESPI::SLAVE0\_INT\_STS)**

Read,Write-1-to-clear. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000070; ESPI=FEC2\_0000h

Bits	Description
31	<b>FLASH_REQ_INT.</b> Read,Write-1-to-clear. Reset: 0. <b>Description:</b> For MAFS, Flash Request Received Status:Upstream Flash Request is received and valid to read. This bit set when Upstream Flash Request is received and valid to read. For SAFS, Flash Completion Received Status:This bit set when Flash completion with data (for flash read), or flash completion without data (for flash write or flash erase) or unsuccessful completion are received. Only for SW programming interface sending PUT_FLASH_NP. The condition is when GET_FLASH_C occurs, or PUT_FLASH_NP with immediately completion response. [PUT_FLASH_NP (write/erase) cycle with header response (PUT_FLASH_NP_HEADER_EN set to 1) or PUT_FLASH_NP (read) cycle with header/data response (PUT_FLASH_NP_HEADER_DATA_EN set to 1)].
30	<b>RXOOB_INT.</b> Read,Write-1-to-clear. Reset: 0. OOB Message Received Status:Upstream OOB Message is received and valid to read. This bit set when Upstream OOB Message is received and valid to read.
29	<b>RXMSG_INT.</b> Read,Write-1-to-clear. Reset: 0. Peripheral Message Received Status:Upstream Peripheral Msg is received and valid to read. This bit set when Upstream Peripheral Msg is received and valid to read.
28	<b>DNCMD_INT.</b> Read,Write-1-to-clear. Reset: 0. Downstream Register Command Complete Status:Downstream eSPI Registers Command has completed. This bit is set when downstream register programming has been completed. Software can program next command or get the data.
27	<b>RXVW_GRP3_INT.</b> Read,Write-1-to-clear. Reset: 0. Virtual Wire Index Group 3 Received Status:Virtual Wire Index Group3 register specified Virtual Wire Packet is received. This bit set when Virtual Wire Index Group3 register specified VW packet has been received.
26	<b>RXVW_GRP2_INT.</b> Read,Write-1-to-clear. Reset: 0. Virtual Wire Index Group 2 Received Status:Virtual Wire Index Group2 register specified Virtual Wire Packet is received. This bit set when Virtual Wire Index Group2 register specified VW packet has been received.
25	<b>RXVW_GRP1_INT.</b> Read,Write-1-to-clear. Reset: 0. Virtual Wire Index Group 1 Received Status:Virtual Wire Index Group1 register specified Virtual Wire Packet is received. This bit set when Virtual Wire Index Group1 register specified VW packet has been received.
24	<b>RXVW_GRP0_INT.</b> Read,Write-1-to-clear. Reset: 0. Virtual Wire Index Group 0 Received Status:Virtual Wire Index Group0 register specified Virtual Wire Packet is received. This bit set when Virtual Wire Index Group0 register specified VW packet has been received.
23:20	Reserved.
19	<b>WDG_TIMEOUT_INT.</b> Read,Write-1-to-clear. Reset: 0. Alink Bus Watch Dog Timer Timeout Status:Alink bus Watch dog timer timeout. This bit set when downstream command retry many times, and over the watch dog timeout value specified in eSPI Global Control and Status Register0
18	<b>MST_ABORT_INT.</b> Read,Write-1-to-clear. Reset: 0. Alink Bus Master Abort Status:eSPI Controller Master aborts. This bit set when eSPI controller doing Master Abort.
17	<b>UPFIFO_WDGT.</b> Read,Write-1-to-clear. Reset: 0. This bit is set when UPFIFO watchdog timer times out, and at the same time, eSPIx10[3], UPFIFO Valid Status bit will be cleared since this's a software error by not reading data out of UPFIFO in time, which might cause Get_Status not to be sent out for the incoming Alert event.
16	Reserved.
15	<b>PROTOCOL_ERR_INT.</b> Read,Write-1-to-clear. Reset: 0. Protocol ERROR detected Status:Protocol ERROR detected.
14	<b>RXFLASH_OVERFLOW_INT.</b> Read,Write-1-to-clear. Reset: 0. FLASH Packet Data Length Over 256 bytes Status:FLASH Packet data Over 256 bytes. This bit set when FLASH packet data length is over 256 bytes.
13	<b>RXMSG_OVERFLOW_INT.</b> Read,Write-1-to-clear. Reset: 0. Peripheral Message Data Length Over 32 bytes Status:Peripheral Message Data Packet data Over 32 bytes. This bit set when Peripheral Message Data length is over 32 bytes.
12	<b>RXOOB_OVERFLOW_INT.</b> Read,Write-1-to-clear. Reset: 0. OOB Packet Data Length Over 256 bytes Status:OOB Packet data Over 256 bytes. This bit set when OOB packet data length is over 256 bytes.

11	<b>ILLEGAL_LEN_INT.</b> Read,Write-1-to-clear. Reset: 0. Illegal Response Length Received Status:Illegal Length Received. This bit set when slave returns a wrong length. Example 1.When peripheral downstream nonpost command with length = 3 receives slave's CPL/CPLD with length not equal to 3. 2.When slave sends the upstream request with length larger than max payload(64 bytes)
10	<b>ILLEGAL_TAG_INT.</b> Read,Write-1-to-clear. Reset: 0. Illegal Response Tag Received Status:Illegal Tag Received. This bit set when slave returns a wrong tag. For example, Peripheral send MEMR with Tag=0, but slave returns CPLD with Tag = 5.
9	<b>UNSUCSS_CPL_INT.</b> Read,Write-1-to-clear. Reset: 0. Unsuccessful CPL Received Status:Unsuccessful completion without data received. This bit will set when Unsuccessful Completion Packet is received.
8	<b>UNKNOWN_CT_INT.</b> Read,Write-1-to-clear. Reset: 0. Invalid Cycle Type Received Status:Unrecognized Cycle Type received. This bit will set when Unrecognized Cycle Type is received.
7	<b>UNKNOWN_RSP_INT.</b> Read,Write-1-to-clear. Reset: 0. Invalid Response Code Received Status:Unrecognized Response received. This bit will set when Unrecognized response code is received.
6	<b>NON_FATAL_ERR_INT.</b> Read,Write-1-to-clear. Reset: 0. NON_FATAL_ERROR Response Code Received Status:NON_FATAL_ERROR Response Code received from slave. This bit will set when NON_FATAL_ERROR response code is received.
5	<b>FATAL_ERR_INT.</b> Read,Write-1-to-clear. Reset: 0. FATAL_ERROR Response Code Received Status:FATAL_ERROR Response Code received from slave. This bit will set when FATAL_ERROR response code is received.
4	<b>NO_RSP_INT.</b> Read,Write-1-to-clear. Reset: 0. NO_RESPONSE Code received Status:No_Response Response Code received from slave. This bit will set when No_Response response code is received.
3	Reserved.
2	<b>CRC_ERR_INT.</b> Read,Write-1-to-clear. Reset: 0. CRC Error detected Status:CRC ERROR detected on response phase. This bit will set when CRC ERROR is detected on response phase.
1	<b>WAIT_TIMEOUT_INT.</b> Read,Write-1-to-clear. Reset: 0. eSPI Bus Wait State Insertion Max Out Status:eSPI Wait State timer timeout. This bit will set when Slave insert more wait state than the counter specified in eSPI Global Control and Status Register0
0	<b>BUS_ERR_INT.</b> Read,Write-1-to-clear. Reset: 0. eSPI Bus Timing Error Status: eSPI Bus timing Error is detected. This bit will set when eSPI link block detect Slave doesn't driver 1 after response CRC, and before CS# de-asserted.

**ESPIx00000074 (FCH::ITF::ESPI::SLAVE0\_RXMSG\_HDR0)**

Read-only. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000074; ESPI=FEC2\_0000h

Bits	Description
31:24	<b>BYTE3.</b> Read-only. Reset: 00h. Received Peripheral Message code
23:16	<b>BYTE2.</b> Read-only. Reset: 00h. Received Peripheral Message Length[7:0]
15:8	<b>BYTE1.</b> Read-only. Reset: 00h. [15:12]: Tag [11:8]: Length[11:8]
7:0	<b>CYCLETYP.</b> Read-only. Reset: 00h. CycleType[7:0] for Peripheral Msg/MsgD

**ESPIx00000078 (FCH::ITF::ESPI::SLAVE0\_RXMSG\_HDR1)**

Read-only. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000078; ESPI=FEC2\_0000h

Bits	Description
31:24	<b>SPECIFIC_BYTE3.</b> Read-only. Reset: 00h. Peripheral Message Specific Byte3
23:16	<b>SPECIFIC_BYTE2.</b> Read-only. Reset: 00h. Peripheral Message Specific Byte2
15:8	<b>SPECIFIC_BYTE1.</b> Read-only. Reset: 00h. Peripheral Message Specific Byte1
7:0	<b>SPECIFIC_BYTE0.</b> Read-only. Reset: 00h. Peripheral Message Specific Byte0

**ESPIx0000007C (FCH::ITF::ESPI::SLAVE0\_RXMSG\_DATA\_PORT)**

Read-only. Reset: 0000\_0000h.

\_aliasHOST; ESPIx0000007C; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>RXMSG_DATA</b> . Read-only. Reset: 0000_0000h.

**ESPIx00000080 (FCH::ITF::ESPI::SLAVE0\_IO\_BASE\_REG2)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000080; ESPI=FEC2\_0000h

Bits	Description
31:16	<b>RANGE5</b> . Read-write. Reset: 0000h. IO decode base address for Range 5
15:0	<b>RANGE4</b> . Read-write. Reset: 0000h. IO decode base address for Range 4

**ESPIx00000084 (FCH::ITF::ESPI::SLAVE0\_IO\_BASE\_REG3)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000084; ESPI=FEC2\_0000h

Bits	Description
31:16	<b>RANGE7</b> . Read-write. Reset: 0000h. IO decode base address for Range 7
15:0	<b>RANGE6</b> . Read-write. Reset: 0000h. IO decode base address for Range 6

**ESPIx00000088 (FCH::ITF::ESPI::SLAVE0\_IO\_SIZE1)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000088; ESPI=FEC2\_0000h

Bits	Description
31:24	<b>RANGE7</b> . Read-write. Reset: 00h. Programmable IO Range7 size
23:16	<b>RANGE6</b> . Read-write. Reset: 00h. Programmable IO Range6 size
15:8	<b>RANGE5</b> . Read-write. Reset: 00h. Programmable IO Range5 size
7:0	<b>RANGE4</b> . Read-write. Reset: 00h. Programmable IO Range4 size

**ESPIx0000008C (FCH::ITF::ESPI::SLAVE0\_IO\_BASE\_REG4)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx0000008C; ESPI=FEC2\_0000h

Bits	Description
31:16	<b>RANGE9</b> . Read-write. Reset: 0000h. IO decode base address for Range 9
15:0	<b>RANGE8</b> . Read-write. Reset: 0000h. IO decode base address for Range 8

**ESPIx00000090 (FCH::ITF::ESPI::SLAVE0\_IO\_BASE\_REG5)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000090; ESPI=FEC2\_0000h

Bits	Description
31:16	<b>RANGE11</b> . Read-write. Reset: 0000h. IO decode base address for Range 11
15:0	<b>RANGE10</b> . Read-write. Reset: 0000h. IO decode base address for Range 10

**ESPIx00000094 (FCH::ITF::ESPI::SLAVE0\_IO\_SIZE2)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000094; ESPI=FEC2\_0000h

Bits	Description
31:24	<b>RANGE11</b> . Read-write. Reset: 00h. Programmable IO Range11 size
23:16	<b>RANGE10</b> . Read-write. Reset: 00h. Programmable IO Range10 size
15:8	<b>RANGE9</b> . Read-write. Reset: 00h. Programmable IO Range9 size
7:0	<b>RANGE8</b> . Read-write. Reset: 00h. Programmable IO Range8 size

**ESPIx00000098 (FCH::ITF::ESPI::RESERVED\_RXMSG\_REG6)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx00000098; ESPI=FEC2\_0000h

Bits	Description
31:24	Reserved.
23:0	<b>VW_IRQ_Inactive_Polarity</b> . Read-write. Reset: 00_0000h. This field is used to change the polarity of default value of Rx VW IRQ status. The default value of Rx VW IRQ is 0h, and a transition of 0 to 1b with this field shall change the default value of Rx VW IRQ to 1b

**ESPIx0000009C (FCH::ITF::ESPI::SLAVE0\_RXVW)**

Reset: 0007\_0C00h.

\_aliasHOST; ESPIx0000009C; ESPI=FEC2\_0000h

Bits	Description
31	<b>CPUTEMP_REQ.</b> Read,Write-1-to-clear. Reset: 0. CPU Temperature Request(CPUTEMP_REQ): sent by Slave to request to read CPU temperature(not from standard spec). Set when received the CPUTEMP_REQ VW
30	<b>RTCTIME_REQ.</b> Read,Write-1-to-clear. Reset: 0. RTC TIME Request(RTCTIME_REQ): sent by Slave to request to read RTC time(not from standard spec). Set when received the RTCTIME_REQ VW
29:20	Reserved.
19	<b>HOST_RST_ACK.</b> Read,Write-1-to-clear. Reset: 0. Host Reset Acknowledge(HOST_RST_ACK): Sent by Slave to acknowledge received HOST_RST_WARN virtual wire.
18	<b>RCIN_B.</b> Read,Write-1-only. Reset: 1. Reset CPU INIT(RCIN#): Send to request CPU reset on behalf of the Keyboard controller.
17	<b>SMI_B.</b> Read,Write-1-only. Reset: 1. System Management Interrupt(SMI#): Sent as general Purpose alert resulting in SMI code being invoked by BIOS.
16	<b>SCI_B.</b> Read,Write-1-only. Reset: 1. System Controller Interrupt(SCI#): Sent as general Purpose alert resulting in ACPI method being invoked by OS.
15	<b>SLAVE_BOOT_LOAD_STS.</b> Read,Write-1-to-clear. Reset: 0. Slave Boot Load Status: Sent by EC or BMC upon completion of Slave Boot Load from the master attached flash. 0:The boot image is corrupted, incomplete or otherwise unusable. 1:The boot code load was successful and that the integrity of the image is intact, or the boot code load from master attached flash is not required.
14	<b>ERROR_NONFATAL.</b> Read,Write-1-to-clear. Reset: 0. NON FATAL ERROR(ERROR_NONFATAL): Sent when a non-fatal error is detected not due to eSPI transaction on the bus.Note: Non-atal Error due to transaction on eSPI bus will be signaled through RSP phase.
13	<b>ERROR_FATAL.</b> Read,Write-1-to-clear. Reset: 0. Fatal Error(ERROR_FATAL): Sent when a fatal error is detected not due to eSPI transaction on the bus.Note: Fatal Error due to transaction on eSPI bus will be signaled through RSP phase.
12	<b>SLAVE_BOOT_LOAD_DONE.</b> Read,Write-1-to-clear. Reset: 0. Slave Boot Load Done(SLAVE_BOOT_LOAD_DONE):Sent when EC or BMC has completed its boot process as indication to eSPI master to continue with the G3 to S0 exit.
11	<b>PME_B.</b> Read,Write-1-only. Reset: 1. PCI Power Management Event(PME#):Shared by multiple eSPI
10	<b>WAKE_B.</b> Read,Write-1-only. Reset: 1. To wavke the Host from Sx on any event(WAKE#)
9	Reserved.
8	<b>OOB_RST_ACK.</b> Read,Write-1-to-clear. Reset: 0. OOB reset Acknowledge(OOB_RST_ACK):Sent by Slave to acknowledge received OOB_RST_ACK virtual wire from Host.
7:5	<b>IRQ_STS.</b> Read-only. Reset: 0h. <b>Description:</b> IRQ Satus: IRQ status which specified by IRQ selection: Bits Status 000 IRQ keep 0 unchanged 001 IRQ keep 1 unchanged 010 IRQ changed from 1 to 0 (Clear) 011 IRQ changed from 0 to 1 (Set) 100 IRQ changed from 0->1->0(High pulse) 101 IRQ changed from 1->0->1(Low pulse) 110 IRQ changed from 1->1->0 or 1->0->0 or 0->0->0 111 IRQ changed from 0->0->1 or 0->1->1 or 1->1->1
4:0	<b>IRQ_SEL.</b> Read-write. Reset: 00h.



	<b>Description:</b> This field determine Slave N Received Virtual Wires Register bit[7:5] output which IRQ status. Bits Selection 00000: IRQ0 00001 IRQ1 .... 10111:IRQ23 others:Reserved
--	---

#### ESPIx000000A0 (FCH::ITF::ESPI::SLAVE0\_RXVW\_DATA)

Read-only. Reset: 0000_0000h.	
_aliasHOST; ESPIx000000A0; ESPI=FEC2_0000h	
Bits	Description
31:24	<b>GRP3.</b> Read-only. Reset: 00h. <b>Description:</b> Group3 Virtual Wire Data Register: When VW MISC CONTRL register bit3 set, eSPI master will check each received VW Index, if the received Index matches with Group3 Virtual Wire Index Selection Register, eSPI Master will update this field with the new received value.
23:16	<b>GRP2.</b> Read-only. Reset: 00h. <b>Description:</b> Group2 Virtual Wire Data Register: When VW MISC CONTRL register bit2 set, eSPI master will check each received VW Index, if the received Index matches with Group2 Virtual Wire Index Selection Register, eSPI Master will update this field with the new received value.
15:8	<b>GRP1.</b> Read-only. Reset: 00h. <b>Description:</b> Group1 Virtual Wire Data Register: When VW MISC CONTRL register bit1 set, eSPI master will check each received VW Index, if the received Index matches with Group1 Virtual Wire Index Selection Register, eSPI Master will update this field with the new received value.
7:0	<b>GRP0.</b> Read-only. Reset: 00h. <b>Description:</b> Group0 Virtual Wire Data Register: When VW MISC CONTRL register bit0 set, eSPI master will check each received VW Index, if the received Index matches with Group0 Virtual Wire Index Selection Register, eSPI Master will update this field with the new received value.

#### ESPIx000000A4 (FCH::ITF::ESPI::SLAVE0\_RXVW\_INDEX)

Read-write. Reset: 0000_0000h.	
_aliasHOST; ESPIx000000A4; ESPI=FEC2_0000h	
Bits	Description
31:24	<b>GRP3.</b> Read-write. Reset: 00h. Group3 Virtual Wire Index Selection Register
23:16	<b>GRP2.</b> Read-write. Reset: 00h. Group2 Virtual Wire Index Selection Register
15:8	<b>GRP1.</b> Read-write. Reset: 00h. Group1 Virtual Wire Index Selection Register
7:0	<b>GRP0.</b> Read-write. Reset: 00h. Group0 Virtual Wire Index Selection Register

**ESPIx000000A8 (FCH::ITF::ESPI::SLAVE0\_RXVW\_MISC\_CNTL)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000A8; ESPI=FEC2\_0000h

Bits	Description
31	<b>IRQ23_MASK</b> . Read-write. Reset: 0.
30	<b>IRQ22_MASK</b> . Read-write. Reset: 0.
29	<b>IRQ21_MASK</b> . Read-write. Reset: 0.
28	<b>IRQ20_MASK</b> . Read-write. Reset: 0.
27	<b>IRQ19_MASK</b> . Read-write. Reset: 0.
26	<b>IRQ18_MASK</b> . Read-write. Reset: 0.
25	<b>IRQ17_MASK</b> . Read-write. Reset: 0.
24	<b>IRQ16_MASK</b> . Read-write. Reset: 0.
23	<b>IRQ15_MASK</b> . Read-write. Reset: 0.
22	<b>IRQ14_MASK</b> . Read-write. Reset: 0.
21	<b>IRQ13_MASK</b> . Read-write. Reset: 0.
20	<b>IRQ12_MASK</b> . Read-write. Reset: 0.
19	<b>IRQ11_MASK</b> . Read-write. Reset: 0.
18	<b>IRQ10_MASK</b> . Read-write. Reset: 0.
17	<b>IRQ9_MASK</b> . Read-write. Reset: 0.
16	<b>IRQ8_MASK</b> . Read-write. Reset: 0.
15	<b>IRQ7_MASK</b> . Read-write. Reset: 0.
14	<b>IRQ6_MASK</b> . Read-write. Reset: 0.
13	<b>IRQ5_MASK</b> . Read-write. Reset: 0.
12	<b>IRQ4_MASK</b> . Read-write. Reset: 0.
11	<b>IRQ3_MASK</b> . Read-write. Reset: 0.
10	<b>IRQ2_MASK</b> . Read-write. Reset: 0.
9	<b>IRQ1_MASK</b> . Read-write. Reset: 0.
8	<b>IRQ0_MASK</b> . Read-write. Reset: 0.
7	<b>CPUTEMP_RTCTIME_VW_INDEX_SEL</b> . Read-write. Reset: 0. CPU_TEMP/RTC TIME VW index selection. 0:index 53(decimal), 1: index 63(decimal)
6	<b>CPUTEMP_RTCTIME_VW_EN</b> . Read-write. Reset: 0. Enabled Hardware to Receive VW packet of index 53(63) from Slave to indicate the slave to request CPU_TEMP/RTC Time info. 1: Enable, 0: Disable.
5	Reserved.
4	<b>SUS_STAT_VWEN</b> . Read-write. Reset: 0. Enabled Hardware to send VW packet when SUS_STAT# changes. 1: Enable, 0: Disable.
3	<b>GRP3_EN</b> . Read-write. Reset: 0. GRP3 Enable : When Set, VW channel will check received Index, if Index is same as Group3 Index register setting; VW will store the Data into Group3 Data register.
2	<b>GRP2_EN</b> . Read-write. Reset: 0. GRP2 Enable : When Set, VW channel will check received Index, if Index is same as Group2 Index register setting; VW will store the Data into Group2 Data register.
1	<b>GRP1_EN</b> . Read-write. Reset: 0. GRP1 Enable : When Set, VW channel will check received Index, if Index is same as Group1 Index register setting; VW will store the Data into Group1 Data register.
0	<b>GRP0_EN</b> . Read-write. Reset: 0. GRP0 Enable : When Set, VW channel will check received Index, if Index is same as Group0 Index register setting; VW will store the Data into Group0 Data register.

**ESPIx000000AC (FCH::ITF::ESPI::SLAVE0\_RXVW\_POLARITY)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000AC; ESPI=FEC2\_0000h

Bits	Description
31:29	Reserved.
28	<b>OOB_RST_WARN_EN.</b> Read-write. Reset: 0. OOB_RST_WARN Enable: Enable Hardware sending Virtual Wire packet when HOST_RST_WARN change. There is no physical oob_Rst_warn output, so if there is HOST_RST_WARN asserted OOB_RST_WARN VW will be send if this bit is set. 1:enable; 0:disable
27	<b>NMIOUT_EN.</b> Read-write. Reset: 0. NMIOUT Enable: Enable Hardware sending Virtual Wire packet when NMIOUT change. 1:enable; 0:disable
26	<b>SMIOUT_EN.</b> Read-write. Reset: 0. SMIOUT Enable: Enable Hardware sending Virtual Wire packet when SMIOUT change. 1:enable; 0:disable
25	<b>HOST_RST_WARN_DIS.</b> Read-write. Reset: 0. HOST_RST_WARN Disable: Default is Enable Hardware sending Virtual Wire packet when HOST_RST_WARN change. 0:enable; 1:disable.
24	<b>PLTRSTB_DIS.</b> Read-write. Reset: 0. PLTRST# Disable: Default 0 to Enable Hardware sending Virtual Wire packet when PLTRST# change. 0:enable; 1:disable.
23	<b>IRQ23_POLARITY.</b> Read-write. Reset: 0.
22	<b>IRQ22_POLARITY.</b> Read-write. Reset: 0.
21	<b>IRQ21_POLARITY.</b> Read-write. Reset: 0.
20	<b>IRQ20_POLARITY.</b> Read-write. Reset: 0.
19	<b>IRQ19_POLARITY.</b> Read-write. Reset: 0.
18	<b>IRQ18_POLARITY.</b> Read-write. Reset: 0.
17	<b>IRQ17_POLARITY.</b> Read-write. Reset: 0.
16	<b>IRQ16_POLARITY.</b> Read-write. Reset: 0.
15	<b>IRQ15_POLARITY.</b> Read-write. Reset: 0.
14	<b>IRQ14_POLARITY.</b> Read-write. Reset: 0.
13	<b>IRQ13_POLARITY.</b> Read-write. Reset: 0.
12	<b>IRQ12_POLARITY.</b> Read-write. Reset: 0.
11	<b>IRQ11_POLARITY.</b> Read-write. Reset: 0.
10	<b>IRQ10_POLARITY.</b> Read-write. Reset: 0.
9	<b>IRQ9_POLARITY.</b> Read-write. Reset: 0.
8	<b>IRQ8_POLARITY.</b> Read-write. Reset: 0.
7	<b>IRQ7_POLARITY.</b> Read-write. Reset: 0.
6	<b>IRQ6_POLARITY.</b> Read-write. Reset: 0.
5	<b>IRQ5_POLARITY.</b> Read-write. Reset: 0.
4	<b>IRQ4_POLARITY.</b> Read-write. Reset: 0.
3	<b>IRQ3_POLARITY.</b> Read-write. Reset: 0.
2	<b>IRQ2_POLARITY.</b> Read-write. Reset: 0.
1	<b>IRQ1_POLARITY.</b> Read-write. Reset: 0.
0	<b>IRQ0_POLARITY.</b> Read-write. Reset: 0.

**ESPIx000000B0 (FCH::ITF::ESPI::SLAVE0\_IO\_BASE\_REG6)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000B0; ESPI=FEC2\_0000h

Bits	Description
31:16	<b>RANGE13.</b> Read-write. Reset: 0000h. IO decode base address for Range 13
15:0	<b>RANGE12.</b> Read-write. Reset: 0000h. IO decode base address for Range 12

**ESPIx000000B4 (FCH::ITF::ESPI::SLAVE0\_IO\_BASE\_REG7)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000B4; ESPI=FEC2\_0000h

Bits	Description
31:16	<b>RANGE15.</b> Read-write. Reset: 0000h. IO decode base address for Range 15
15:0	<b>RANGE14.</b> Read-write. Reset: 0000h. IO decode base address for Range 14

**ESPIx000000B8 (FCH::ITF::ESPI::SLAVE0\_IO\_SIZE3)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000B8; ESPI=FEC2\_0000h

Bits	Description
31:24	<b>RANGE15.</b> Read-write. Reset: 00h. Programmable IO Range15 size
23:16	<b>RANGE14.</b> Read-write. Reset: 00h. Programmable IO Range14 size
15:8	<b>RANGE13.</b> Read-write. Reset: 00h. Programmable IO Range13 size
7:0	<b>RANGE12.</b> Read-write. Reset: 00h. Programmable IO Range12 size

**ESPIx000000BC (FCH::ITF::ESPI::SLAVE0\_MMIO\_BASE\_REG4)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000BC; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>RANGE4.</b> Read-write. Reset: 0000_0000h. MMIO decode base address for Range 4

**ESPIx000000C0 (FCH::ITF::ESPI::SLAVE0\_MMIO\_SIZE\_REG2)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000C0; ESPI=FEC2\_0000h

Bits	Description
31:25	Reserved.
24:0	<b>RANGE4.</b> Read-write. Reset: 000_0000h. Programmable MMIO Range4 size, 1M - vA.6, 32M - vA.7

**ESPIx000000C4 (FCH::ITF::ESPI::MMIO\_CPUTEMP)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000C4; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>MMIO_CPUTEMP.</b> Read-write. Reset: 0000_0000h. MMIO address to send down CPU temperture.

**ESPIx000000C8 (FCH::ITF::ESPI::MMIO\_RTCTIME)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000C8; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>MMIO_RTCTIME.</b> Read-write. Reset: 0000_0000h. MMIO address to send down RTC TIME.

**ESPIx000000CC (FCH::ITF::ESPI::ESPI\_MiscCtrl1)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000CC; ESPI=FEC2\_0000h

Bits	Description
31:17	<b>ESPI_MiscCtrl1_RESERVED0.</b> Read-write. Reset: 0000h.
16	<b>UPFIFO_WDG_TimerEn.</b> Read-write. Reset: 0. Set the bit to enable the UPFIFO watchdog timer.
15:0	<b>UPFIFO_WDG_Timer.</b> Read-write. Reset: 0000h.  <b>Description:</b> UPFIFO Watch dog counter: This counter times the duration of eSPIx10[3] staying at high level in case software is too slow to clear eSPIx10[3]. This counter will cumulate itself by 1 every 16 cycles of Alink clock. The setting value for this field is recommended to be less than the Alink Watchdog Timer so that the below situation can be saved back. In some circumstances, software might be very slow or fail to clear eSPIx10[3]. At that time, if a Put_Flash_NP from register interface is deferred when SAFS flash modifier is enabled, eSPIx10[3] staying at high will prevent controller from sending out Get_Status when Alert event is received (Otherwise, device might respond with flash completion data together with Accept response for Get_Status, which will override the UPFIFO). When this counter times out, it will clear eSPIx10[3] and record this software error into eSPIx70[17].

**ESPIx000000D0 (FCH::ITF::ESPI::ESPI\_LOWSECURITY\_CONTROL\_REG\_RESERVED1)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000D0; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>ESPI_LOWSECURITY_CONTROL_REG_RESERVED1.</b> Read-write. Reset: 0000_0000h.

**ESPIx000000D4 (FCH::ITF::ESPI::ESPI\_LOWSECURITY\_CONTROL\_REG\_RESERVED2)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000D4; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>ESPI_LOWSECURITY_CONTROL_REG_RESERVED2.</b> Read-write. Reset: 0000_0000h.

**ESPIx000000D8 (FCH::ITF::ESPI::ESPI\_LOWSECURITY\_CONTROL\_REG\_RESERVED3)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000D8; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>ESPI_LOWSECURITY_CONTROL_REG_RESERVED3.</b> Read-write. Reset: 0000_0000h.

**ESPIx000000DC (FCH::ITF::ESPI::ESPI\_LOWSECURITY\_CONTROL\_REG\_RESERVED4)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000DC; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>ESPI_LOWSECURITY_CONTROL_REG_RESERVED4.</b> Read-write. Reset: 0000_0000h.

**ESPIx000000E0 (FCH::ITF::ESPI::ESPI\_LOWSECURITY\_CONTROL\_REG\_RESERVED5)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000E0; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>ESPI_LOWSECURITY_CONTROL_REG_RESERVED5.</b> Read-write. Reset: 0000_0000h.

**ESPIx000000E4 (FCH::ITF::ESPI::ESPI\_LOWSECURITY\_CONTROL\_REG\_RESERVED6)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000E4; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>ESPI_LOWSECURITY_CONTROL_REG_RESERVED6.</b> Read-write. Reset: 0000_0000h.

**ESPIx000000E8 (FCH::ITF::ESPI::ESPI\_LOWSECURITY\_CONTROL\_REG\_RESERVED7)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000E8; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>ESPI_LOWSECURITY_CONTROL_REG_RESERVED7</b> . Read-write. Reset: 0000_0000h.

**ESPIx000000EC (FCH::ITF::ESPI::ESPI\_LOWSECURITY\_CONTROL\_REG\_RESERVED8)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000EC; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>ESPI_LOWSECURITY_CONTROL_REG_RESERVED8</b> . Read-write. Reset: 0000_0000h.

**ESPIx000000F0 (FCH::ITF::ESPI::ESPI\_LOWSECURITY\_CONTROL\_REG\_RESERVED9)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000F0; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>ESPI_LOWSECURITY_CONTROL_REG_RESERVED9</b> . Read-write. Reset: 0000_0000h.

**ESPIx000000F4 (FCH::ITF::ESPI::ESPI\_LOWSECURITY\_CONTROL\_REG\_RESERVED10)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000F4; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>ESPI_LOWSECURITY_CONTROL_REG_RESERVED10</b> . Read-write. Reset: 0000_0000h.

**ESPIx000000F8 (FCH::ITF::ESPI::ESPI\_LOWSECURITY\_CONTROL\_REG\_RESERVED11)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000F8; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>ESPI_LOWSECURITY_CONTROL_REG_RESERVED11</b> . Read-write. Reset: 0000_0000h.

**ESPIx000000FC (FCH::ITF::ESPI::ESPI\_LOWSECURITY\_CONTROL\_REG\_RESERVED12)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; ESPIx000000FC; ESPI=FEC2\_0000h

Bits	Description
31:0	<b>ESPI_LOWSECURITY_CONTROL_REG_RESERVED12</b> . Read-write. Reset: 0000_0000h.

**7.3.9 MISC Registers****7.3.9.1 Miscellaneous (MISC) Registers**

**MISCx00000000 (FCH::MISC::gppclkcontrol)**

Read-write. Reset: 0000\_3FFFh.

\_aliasHOST; MISCx00000000; MISC=FED8\_0E00h

Bits	Description
31:14	Reserved.
13:12	<b>gpp_clk6_clock_request_mapping.</b> Read-write. Reset: 3h. <b>Description:</b> GPP6 PCIE clock pins (GPP_CLK6P/GPP_CLK6N) output control by CLKREQ6# pin GPP_CLK6P/GPP_CLK6N pins are powered off when CHIP is strapped to use an external clock, and powered on when CHIP is strapped to operate in integrated clock mode. When CHIP is in integrated clock mode, GPP6 PCIE clock can be powered off according to the CLK_REQ mapping table below, and the selected CLK_REQ6# input can power off the GPP6 PCIE clock output pins if it is asserted. GPP6_CLKREQ_Mapping: 00 Off 01 CLK_REQ6# 10 Off, reserved 11 On (default)
11:10	<b>gpp_clk5_clock_request_mapping.</b> Read-write. Reset: 3h. <b>Description:</b> GPP5 PCIE clock pins (GPP_CLK5P/GPP_CLK5N) output control by CLKREQ5# (or CLKREQG#) pin GPP_CLK5P/GPP_CLK5N pins are powered off when CHIP is strapped to use an external clock, and powered on when CHIP is strapped to operate in integrated clock mode. When CHIP is in integrated clock mode, GPP5 PCIE clock can be powered off according to the CLK_REQ mapping table below, and the selected CLK_REQ5# input can power off the GPP5 PCIE clock output pins if it is asserted. GPP5_CLKREQ_Mapping: 00 Off 01 CLK_REQ5# 10 Off, reserved 11 On (default)
9:8	<b>gpp_clk3_clock_request_mapping.</b> Read-write. Reset: 3h. <b>Description:</b> GPP3 PCIE clock pins (GPP_CLK3P/GPP_CLK3N) output control by CLKREQ3# pin GPP_CLK3P/GPP_CLK3N pins are powered off when CHIP is strapped to use an external clock, and powered on when CHIP is strapped to operate in integrated clock mode. When CHIP is in integrated clock mode, GPP3 PCIE clock can be powered off according to the CLK_REQ mapping table below, and the selected CLK_REQ3# input can power off the GPP3 PCIE clock output pins if it is asserted. GPP3_CLKREQ_Mapping: 00 Off 01 CLK_REQ3# 10 Off, reserved 11 On (default)
7:6	<b>gpp_clk2_clock_request_mapping.</b> Read-write. Reset: 3h. <b>Description:</b> GPP2 PCIE clock pins (GPP_CLK2P/GPP_CLK2N) output control by CLKREQ2# pin GPP_CLK2P/GPP_CLK2N pins are powered off when CHIP is strapped to use an external clock, and powered on when CHIP is strapped to operate in integrated clock mode. When CHIP is in integrated clock mode, GPP2 PCIE clock can be powered off according to the CLK_REQ mapping table below, and the selected CLK_REQ2# input can power off the GPP2 PCIE clock output pins if it is asserted. GPP2_CLKREQ_Mapping: 00 Off 01 CLK_REQ2# 10 Off, reserved 11 On (default)
5:4	<b>gpp_clk4_clock_request_mapping.</b> Read-write. Reset: 3h.



	<p><b>Description:</b> GPP4 PCIE clock pins (GPP_CLK4P/GPP_CLK4N) output control by CLKREQ4# pin GPP_CLK4P/GPP_CLK4N pins are powered off when CHIP is strapped to use an external clock, and powered on when CHIP is strapped to operate in integrated clock mode. When CHIP is in integrated clock mode, GPP4 PCIE clock can be powered off according to the CLK_REQ mapping table below, and the selected CLK_REQ4# input can power off the GPP4 PCIE clock output pins if it is asserted. GPP4_CLKREQ_Mapping: 00 Off 01 CLK_REQ4# 10 Off, reserved 11 On (default)</p>
3:2	<p><b>gpp_clk1_clock_request_mapping.</b> Read-write. Reset: 3h. <b>Description:</b> GPP1 PCIE clock pins (GPP_CLK1P/GPP_CLK1N) output control by CLKREQ1# pin GPP_CLK1P/GPP_CLK1N pins are powered off when CHIP is strapped to use an external clock, and powered on when CHIP is strapped to operate in integrated clock mode. When CHIP is in integrated clock mode, GPP1 PCIE clock can be powered off according to the CLK_REQ mapping table below, and the selected CLK_REQ1# input can power off the GPP1 PCIE clock output pins if it is asserted. GPP1_CLKREQ_Mapping: 00 Off 01 CLK_REQ1# 10 Off, reserved 11 On (default)</p>
1:0	<p><b>gpp_clk0_clock_request_mapping.</b> Read-write. Reset: 3h. <b>Description:</b> GPP0 PCIE clock pins (GPP_CLK0P/GPP_CLK0N) output control by CLKREQ0# pin GPP_CLK0P/GPP_CLK0N pins are powered off when CHIP is strapped to use an external clock, and powered on when CHIP is strapped to operate in integrated clock mode. When CHIP is in integrated clock mode, GPP0 PCIE clock can be powered off according to the CLK_REQ mapping table below, and the selected CLK_REQ0# input can power off the GPP0 PCIE clock output pins if it is asserted. GPP0_CLKREQ_Mapping: 00 Off 01 CLK_REQ0# 10 Off, reserved 11 On (default)</p>

#### MISCx00000004 (FCH::MISC::clkoutputcntrl)

Reset: 0000\_0000h.

\_aliasHOST; MISCx00000004; MISC=FED8\_0E00h

Bits	Description
31:0	Reserved.



**MISCx00000008 (FCH::MISC::cgpllconfig1)**

Read-write. Reset: 6000\_0000h.

\_aliasHOST; MISCx00000008; MISC=FED8\_0E00h

Bits	Description
31	<b>xtal_refclk2x_clkenb.</b> Read-write. Reset: 0. <b>Description:</b> XTAL_PAD Output REFCLK2X (p6M) clock enableB/disableB 0: Turn on 1: Turn off
30	<b>xtal_clkgen_s5_clken.</b> Read-write. Reset: 1. <b>Description:</b> XTAL_PAD Output CLKGEN_S5 RefClk Enable/disable 0: Turn off 1: Turn on
29	<b>xtal_clkgen_s0_clken.</b> Read-write. Reset: 1. <b>Description:</b> XTAL_PAD Output CLKGEN_S0 RefClk Enable/disable 0: Turn off 48MHz for CLKGEN_S0 by ZSC when in Z10/Z10+. 1: Turn on, free running 48MHz
28:9	Reserved.
8	<b>cg1_refdivsrc_override.</b> Read-write. Reset: 0. <b>Description:</b> CG1PLL Refclk Source Select Override This bit is used to override CG1PLL refclk source default selection. By default, CG1PLL refclk source is from CG_XTAL 48MHz in iCLK mode and 100MHz from external clock chip in eCLK mode. Note: Whether need to apply a reset after set this bit or not depend on x08[4]'s value.
7:5	Reserved.
4	<b>cgpll_refclk_source_update_scheme.</b> Read-write. Reset: 0. <b>Description:</b> CGPLL Refclk Source Update Scheme For CGPLL refclk source update scheme, it can be from either CG_XTAL/CG_PLL generated or external reference source. Note: should get it from ACPI main register instead of from shadow register 0: Switch refclk source on-the-fly 1: Need to apply a PllRstB for switching different refclk source
3:1	Reserved.
0	<b>cg1_spread_spectrum_enable.</b> Read-write. Reset: 0. <b>Description:</b> CG1_PLL Spread Spectrum Enable 0: Disable Spread Spectrum (default) 1: Enable Spread Spectrum

**MISCx0000000C (FCH::MISC::cgpllconfig2)**

Reset: 0000\_0000h.

\_aliasHOST; MISCx0000000C; MISC=FED8\_0E00h

Bits	Description
31:0	Reserved.

**MISCx00000010 (FCH::MISC::cgpllconfig3)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx00000010; MISC=FED8\_0E00h

Bits	Description
31:30	Reserved.
29	<b>cg1pll_fracn_en_override</b> . Read-write. Reset: 0. Used with CG1PLL frac-N and SSC clocking only.
28:13	Reserved.
12:4	<b>cg1pll_fcw0_int_override</b> . Read-write. Reset: 000h. CG1PLL Override: Integer portion of Frequency Control Word0 (a.k.a. feedback divisor0).
3:2	Reserved.
1:0	<b>cg1pll_refclk_div_override</b> . Read-write. Reset: 0h. <b>Description:</b> CG1PLL Override: Reference clock divisor. Settings: b00=1 b01=2 b1x=4

**MISCx00000014 (FCH::MISC::cgpllconfig4)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx00000014; MISC=FED8\_0E00h

Bits	Description
31:24	Reserved.
23:8	<b>cg1pll_fcw1_frac_override</b> . Read-write. Reset: 0000h. CG1PLL Override: Fractional portion of Frequency Control Word1 (a.k.a. feedback divisor1). Intended to be used with frequency ramping. Also used to step PLL frequency and phase for debug.
7:0	Reserved.

**MISCx00000018 (FCH::MISC::cgpllconfig5)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx00000018; MISC=FED8\_0E00h

Bits	Description
31:16	<b>cg1pll_fcw_slew_frac_override</b> . Read-write. Reset: 0000h. <b>Description:</b> CG1PLL Override: Sets SSC freq ramp rate. Set fractional change in programmed frequency per refclk cycle. e.g. 0.5% downspread SSC at 33.3KHz and fbdiv=80. FCW_slewrates_frac = $216 * 0.00485 * 80 / (15\mu s / 10ns) = 17$ Need 31.5KHz and -0.375%
15:0	Reserved.

**MISCx0000001C (FCH::MISC::cgpllconfig6)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx0000001C; MISC=FED8\_0E00h

Bits	Description
31:28	Reserved.
27:26	<b>cgpll_gi_coarse_mant_override</b> . Read-write. Reset: 0h.
25:21	Reserved.
20:17	<b>cg1pll_gp_coarse_exp_override</b> . Read-write. Reset: 0h. CG1PLL Override: Coarse proportional path fp mult exponent
16:13	<b>cg1pll_gp_coarse_mant_override</b> . Read-write. Reset: 0h. CG1PLL Override: Coarse proportional path fp mult mantissa
12:9	<b>cg1pll_gi_coarse_exp_override</b> . Read-write. Reset: 0h. CG1PLL Override: Coarse integral path fp mult exponent
8:0	Reserved.

**MISCx00000020 (FCH::MISC::impcalibration)**

Reset: 0000\_0000h.

\_aliasHOST; MISCx00000020; MISC=FED8\_0E00h

Bits	Description
31:0	Reserved.

**MISCx00000024 (FCH::MISC::clkdrvsth1)**

Read-write. Reset: 2024\_0249h.

\_aliasHOST; MISCx00000024; MISC=FED8\_0E00h

Bits	Description
31:29	<b>gpp6_clock_buffer_driving_strength_control.</b> Read-write. Reset: 1h. <b>Description:</b> Drive Strength control for GPP6 differential Clock Buffers Drive strength addition/subtraction relative to IMP_CTRL[4:0]. bx00 : ~ -10% bx01 : no change from IMP_CTRL bx10 : ~ +10% bx11 : ~ + 20% Default: b001
28:24	Reserved.
23:21	<b>gpp5_clock_buffer_driving_strength_control.</b> Read-write. Reset: 1h. <b>Description:</b> Drive Strength control for GPP5 differential Clock Buffers Drive strength addition/subtraction relative to IMP_CTRL[4:0]. bx00 : ~ -10% bx01 : no change from IMP_CTRL bx10 : ~ +10% bx11 : ~ + 20% Default: b001
20:18	<b>gpp4_clock_buffer_driving_strength_control.</b> Read-write. Reset: 1h. <b>Description:</b> Drive Strength control for GPP4 differential Clock Buffers Drive strength addition/subtraction relative to IMP_CTRL[4:0]. bx00 : ~ -10% bx01 : no change from IMP_CTRL bx10 : ~ +10% bx11 : ~ + 20% Default: b001
17:12	Reserved.
11:9	<b>gpp_clk3_clock_buffer_driving_strength_control.</b> Read-write. Reset: 1h. <b>Description:</b> Drive Strength control for GPP_CLK_3 differential Clock Buffers Drive strength addition/subtraction relative to IMP_CTRL[4:0]. bx00 : ~ -10% bx01 : no change from IMP_CTRL bx10 : ~ +10% bx11 : ~ + 20% Default: b001
8:6	<b>gpp_clk2_clock_buffer_driving_strength_control.</b> Read-write. Reset: 1h. <b>Description:</b> Drive Strength control for GPP_CLK_2 differential Clock Buffers Drive strength addition/subtraction relative to IMP_CTRL[4:0]. bx00 : ~ -10% bx01 : no change from IMP_CTRL bx10 : ~ +10% bx11 : ~ + 20% Default: b001
5:3	<b>gpp_clk1_clock_buffer_driving_strength_control.</b> Read-write. Reset: 1h.

	<b>Description:</b> Drive Strength control for GPP_CLK_1 differential Clock Buffers Drive strength addition/subtraction relative to IMP_CTRL[4:0]. bx00 : ~ -10% bx01 : no change from IMP_CTRL bx10 : ~ +10% bx11 : ~ + 20% Default: b001
2:0	<b>gpp_clk0_clock_buffer_driving_strength_control.</b> Read-write. Reset: 1h. <b>Description:</b> Drive Strength control for GPP_CLK_0 differential Clock Buffers Drive strength addition/subtraction relative to IMP_CTRL[4:0]. bx00 : ~ -10% bx01 : no change from IMP_CTRL bx10 : ~ +10% bx11 : ~ + 20% Default: b001

**MISCx00000028 (FCH::MISC::clkdrvsth2)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx00000028; MISC=FED8\_0E00h

Bits	Description
31:25	<b>debug_sig_sel_3.</b> Read-write. Reset: 00h. <b>Description:</b> Debug Signal Selection 3 These bits are used to select internal signal for debug observation thru uPAD_S5_AGPI03/CLK_48_24_2 pin with xBC[30]=1
24:18	<b>debug_sig_sel_2.</b> Read-write. Reset: 00h. <b>Description:</b> Debug Signal Selection 2 These bits are used to select internal signal for debug observation thru BP_LL_B_L/uPAD_S5_GPIO12/CLK_48_24_1 pin with xBC[29]=1
17:11	<b>debug_sig_sel_1.</b> Read-write. Reset: 00h. <b>Description:</b> Debug Signal Selection 1 These bits are used to select internal signal for debug observation thru uPAD_S5_AGPI09/CLK_48_24_0 pin with xBC[28]=1
10	<b>xtal_pad_pwdn_static_cntl.</b> Read-write. Reset: 0. <b>Description:</b> XTAL_PAD PWDN Static Cntl XTAL_PAD can be turn on/off by program this bit with x2C[24]=0. 0: Turn on XTAL_PAD 1: Trun off XTAL_PAD
9	Reserved.
8	<b>clkb_dp_refclk_selection.</b> Read-write. Reset: 0. <b>Description:</b> CLKB_DP Refclk Selection 0 = 24 MHz, divided by 2 from CG_XTAL (default) 1 = 48 MHz, buffered version from CG_XTAL
7	<b>clkb_dp_refclk_driver_pwdn.</b> Read-write. Reset: 0. <b>Description:</b> CLKB_DP Refclk Driver PWDN 0 = Turn on 1 = Turn off
6:5	Reserved.
4	<b>acp_pll_refclk_refclk_driver_pwdn.</b> Read-write. Reset: 0. <b>Description:</b> ACP_PLL_REFCLK Refclk Driver PWDN 0 = Turn on 1 = Turn off
3:0	Reserved.

**MISCx0000002C (FCH::MISC::clkgatedcntl)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx0000002C; MISC=FED8\_0E00h

Bits	Description
31:22	Reserved.
21	<b>xtal_pad_s5_turnoff_en.</b> Read-write. Reset: 0. <b>Description:</b> XTAL_PAD Turn-Off Enable when at S5 state Set this bit to "1" to allow XTAL_PAD can be turn off when at S5 state if turn-off condition meet (no ACP, No MP2, Not initial G3-to-S5, and no wlan_clk_req).
20	<b>xtal_pad_s3_turnoff_en.</b> Read-write. Reset: 0. <b>Description:</b> XTAL_PAD Turn-Off Enable when at S3 state Set this bit to "1" to allow XTAL_PAD can be turn off when at S3 state if turn-off condition meet (no ACP, No MP2, Not initial G3-to-S5, and no wlan_clk_req).
19	<b>xtal_pad_s0i3_turnoff_en.</b> Read-write. Reset: 0. <b>Description:</b> XTAL_PAD Turn-Off Enable when at S0i3 state Set this bit to "1" to allow XTAL_PAD can be turn off when at S0i3 state if turn-off condition meet (no ACP, No MP2, Not initial G3-to-S5, and no wlan_clk_req).
18	<b>xtal_pad_pwdn_wlanclkreq_maskoff.</b> Read-write. Reset: 0. <b>Description:</b> XTAL_PAD_PWDN Wlan_clk_req MaskOff Set this bit to "1" to allow XTAL_PAD can be turn off regardless Wlan_clk_req status in sleep state. In this situation, register x2C[22] represent wlan_clk_req status (debug purpose).
17	<b>blinkclk_gateoffen.</b> Read-write. Reset: 0. Check: 1. <b>Description:</b> B-Link Clock Gate-Off Enable Internal B-Link clock has two clock trees: one is a free running clock and the other is a gated clock. When all controllers agree to stop the gated B-Link clock and this bit got set, clkgating logic will gate off the clock tree from clock root. Note: Need to enable with PM04[16]=1 (non-sticky) 0: Disable B-Link Clock Gate-Off function. Default 1: Enable B-Link Clock Gate-Off function
16	<b>alinkclk_gateoffen.</b> Read-write. Reset: 0. Check: 1. <b>Description:</b> A-Link Clock Gate-Off Enable Internal A-Link clock has two clock trees: one is a free-running clock and the other is a gated clock. When all controllers agree to stop the gated A-Link clock and this bit got set, clkgating logic will gate off the clock tree from clock root. Note: Need to enable with PM04[16]=1 (non-sticky) 0: Disable A-Link Clock Gate-Off function. Default 1: Enable A-Link Clock Gate-Off function
15:0	Reserved.

**MISCx00000030 (FCH::MISC::cgpll\_configuration0)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx00000030; MISC=FED8\_0E00h

Bits	Description
31:25	<b>debug_sig_sel_4.</b> Read-write. Reset: 00h. <b>Description:</b> Debug Signal Selection 4 These bits are used to select internal signal for debug observation thru uPAD_S5_AGPIO40 pin with xBC[31]=1
24:18	<b>debug_sig_sel_0.</b> Read-write. Reset: 00h. <b>Description:</b> Debug Signal Selection 0 These bits are used to select internal signal for debug observation thru uPAD_S5_X48MO pin with xBC[1]=1
17	<b>uart4_14_77mhz_sclk_enable.</b> Read-write. Reset: 0. <b>Description:</b> UART4 14.77MHz SCLK Enable 0: Turn off UART4 14.77MHz SCLK 1: Turn on UART4 14.77MHz SCLK
16	<b>uart3_14_77mhz_sclk_enable.</b> Read-write. Reset: 0. <b>Description:</b> UART3 14.77MHz SCLK Enable 0: Turn off UART3 14.77MHz SCLK 1: Turn on UART3 14.77MHz SCLK
15	<b>uart2_14_77mhz_sclk_enable.</b> Read-write. Reset: 0. <b>Description:</b> UART2 14.77MHz SCLK Enable 0: Turn off UART2 14.77MHz SCLK 1: Turn on UART2 14.77MHz SCLK
14	<b>uart1_14_77mhz_sclk_enable.</b> Read-write. Reset: 0. <b>Description:</b> UART1 14.77MHz SCLK Enable 0: Turn off UART1 14.77MHz SCLK 1: Turn on UART1 14.77MHz SCLK
13	<b>uart0_14_77mhz_sclk_enable.</b> Read-write. Reset: 0. <b>Description:</b> UART0 14.77MHz SCLK Enable 0: Turn off UART0 14.77MHz SCLK 1: Turn on UART0 14.77MHz SCLK
12	<b>usb_phy_cmlclk_s0i21_dis.</b> Read-write. Reset: 0. <b>Description:</b> USB PHY CML Clock Turn Off at S0i21 State 0: USB PHY CML Clock will not be turn off at S0i21 state 1: USB PHY CML Clock will be turn off at S0i21 state
11	<b>usb_phy_cmlclk_static_off.</b> Read-write. Reset: 0. <b>Description:</b> USB PHY CML Clock Static Turn Off 0: USB PHY CML Clock will not be static turn off 1: USB PHY CML Clock will be static turn off
10	<b>usb_phy_cmlclk_s5_dis.</b> Read-write. Reset: 0. <b>Description:</b> USB PHY CML Clock Turn Off at S5 State 0: USB PHY CML Clock will not be turn off at S5 state 1: USB PHY CML Clock will be turn off at S5 state
9	<b>usb_phy_cmlclk_s0i3_dis.</b> Read-write. Reset: 0. <b>Description:</b> USB PHY CML Clock Turn Off at S0i3 State 0: USB PHY CML Clock will not be turn off at S0i3 state 1: USB PHY CML Clock will be turn off at S0i3 state
8	<b>usb_phy_cmlclk_s3_dis.</b> Read-write. Reset: 0. <b>Description:</b> USB PHY CML Clock Turn Off at S3 State 0: USB PHY CML Clock will not be turn off at S3 state 1: USB PHY CML Clock will be turn off at S3 state



7:4	Reserved.
3	<b>usb0_32k_clk_static_off.</b> Read-write. Reset: 0. <b>Description:</b> USB0 32K CLK Output Static Off 0: Static turn On USB0 32K CLK 1: Static turn Off USB0 32K CLK
2	Reserved.
1	<b>usb_phy_cmlclk_zstate_dis.</b> Read-write. Reset: 0. <b>Description:</b> USB PHY CML Clock Turn Off at Z State 0: USB PHY CML Clock will not be turn off at Z state 1: USB PHY CML Clock will be turn off at Z state
0	Reserved.

**MISCx00000034 (FCH::MISC::cgpll\_configuration1)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx00000034; MISC=FED8\_0E00h

Bits	Description
31:23	<b>cg1pll_fcw1_int_override.</b> Read-write. Reset: 000h. CG1PLL Override: Integer portion of Frequency Control Word0 (a.k.a. feedback divisor1). Intended to be used with frequency ramping. Also used to step PLL frequency and phase for debug.
22:0	Reserved.

**MISCx00000038 (FCH::MISC::cgpll\_configuration2)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx00000038; MISC=FED8\_0E00h

Bits	Description
31:27	Reserved.
26	<b>usb_hc2_clk_switching_mechanism.</b> Read-write. Reset: 0. <b>Description:</b> USB HC2 Clock Switching_ Mechanism USB HC2 Clock can be either 60M generated from CGPLL or 48M generated from XTAL_PAD when CGPLL 60M generated clock is not available. X38[26] is used to select the switching mechanism. 0: HW dynamic switching between 60M and 48M according whether CGPLL is available or not. 1: FW switch thru x38[24]
25	Reserved.
24	<b>usb_hc2_clk_selection.</b> Read-write. Reset: 0. <b>Description:</b> USB HC2 Clock Selection USB HC2 Clock can be either 60M generated from CGPLL or 48M generated from XTAL_PAD, default is 300M. FW can program this bit to switch host clock between 60M and 48M when enter/exit S0i2.1. Please refer USB's requirement. (pair-up with USB's sequence) USB HC2 Clock can be either 60M generated from CGPLL or 48M generated from XTAL_PAD when CGPLL 60M generated clock is not available. X38[26] is used to select the switching mechanism. X38[26] =0, HW dynamic switching between 60M and 48M according whether CGPLL is available or not. X38[26]=1, FW switch thru x38[24] 0: 60MHz generated by CGPLL 1: 48MHz generated from XTAL_PAD
23	<b>usb1_host_clk_switching_mechanism.</b> Read-write. Reset: 0. <b>Description:</b> USB1 Host Clock Switching_ Mechanism USB1 Host Clock can be either 300M generated from CGPLL or 48M generated from XTAL_PAD when CGPLL 300M generated clock is not available. X38[23] is used to select the switching mechanism. 0: HW dynamic switching between 300M and 48M according whether CGPLL is available or not. 1: FW switch thru x38[21]
22	<b>usb7_host_clk_staticoff.</b> Read-write. Reset: 0. <b>Description:</b> USB7 Host Clock Static Off If USB7 is not used in a given platform, USB7_Host_Clk can be turn off by set this bit to "1". 0: Static Off function is not enabled 1: Static Off function is enabled
21	<b>usb1_host_clk_selection.</b> Read-write. Reset: 0. <b>Description:</b> USB1 Host Clock Selection USB1 Host Clock can be either 300M generated from CGPLL or 48M generated from XTAL_PAD, default is 300M. FW can program this bit to switch host clock between 300M and 48M when enter/exit S0i2.1. Please refer USB's requirement. (pair-up with USB's sequence) USB1 Host Clock can be either 300M generated from CGPLL or 48M generated from XTAL_PAD when CGPLL 300M generated clock is not available. X38[23] is used to select the switching mechanism. X38[23] =0, HW dynamic switching between 300M and 48M according whether CGPLL is available or not. X38[23]=1, FW switch thru x38[21] 0: 300MHz generated by CGPLL 1: 48MHz generated from XTAL_PAD
20	<b>usb0_host_clk_switching_mechanism.</b> Read-write. Reset: 0.

	<b>Description:</b> USB0 Host Clock Switching_ Mechanism USB0 Host Clock can be either 300M generated from CGPLL or 48M generated from XTAL_PAD when CGPLL 300M generated clock is not available. X38[20] is used to select the switching mechanism. 0: HW dynamic switching between 300M and 48M according whether CGPLL is available or not. 1: FW switch thru x38[18]
19	<b>usb7_host_clk_selection.</b> Read-write. Reset: 0. <b>Description:</b> USB7 Host Clock Selection USB7 Host Clock can be either 300M generated from CGPLL or 48M generated from XTAL_PAD, default is 300M. FW can program this bit to switch host clock between 300M and 48M when enter/exit S0i2.1. Please refer USB's requirement. (pair-up with USB's sequence) USB7 Host Clock can be either 300M generated from CGPLL or 48M generated from XTAL_PAD when CGPLL 300M generated clock is not available. X38[12] is used to select the switching mechanism. X38[12] =0, HW dynamic switching between 300M and 48M according whether CGPLL is available or not. X38[12]=1, FW switch thru x38[19] 0: 300MHz generated by CGPLL 1: 48MHz generated from XTAL_PAD
18	<b>usb0_host_clk_selection.</b> Read-write. Reset: 0. <b>Description:</b> USB0 Host Clock Selection USB0 Host Clock can be either 300M generated from CGPLL or 48M generated from XTAL_PAD, default is 300M. FW can program this bit to switch host clock between 300M and 48M when enter/exit S0i2.1. Please refer USB's requirement. (pair-up with USB's sequence) USB0 Host Clock can be either 300M generated from CGPLL or 48M generated from XTAL_PAD when CGPLL 300M generated clock is not available. X38[20] is used to select the switching mechanism. X38[20] =0, HW dynamic switching between 300M and 48M according whether CGPLL is available or not. X38[20]=1, FW switch thru x38[18] 0: 300MHz generated by CGPLL 1: 48MHz generated from XTAL_PAD
17:13	Reserved.
12	<b>usb7_host_clk_switching_mechanism.</b> Read-write. Reset: 0. <b>Description:</b> USB7 Host Clock Switching_ Mechanism USB7 Host Clock can be either 300M generated from CGPLL or 48M generated from XTAL_PAD when CGPLL 300M generated clock is not available. X38[12] is used to select the switching mechanism. 0: HW dynamic switching between 300M and 48M according whether CGPLL is available or not. 1: FW switch thru x38[19]
11	<b>usb2_host_clk_staticoff.</b> Read-write. Reset: 0. <b>Description:</b> USB2 Host Clock Static Off If USB2 is not used in a given platform, USB2_Host_Clk can be turn off by set this bit to "1". 0: Static Off function is not enabled 1: Static Off function is enabled
10	<b>usb2_host_clk_selection.</b> Read-write. Reset: 0.

	<b>Description:</b> USB2 Host Clock Selection USB2 Host Clock can be either 300M generated from CGPLL or 48M generated from XTAL_PAD, default is 300M. FW can program this bit to switch host clock between 300M and 48M when enter/exit S0i2.1. Please refer USB's requirement. (pair-up with USB's sequence) USB2 Host Clock can be either 300M generated from CGPLL or 48M generated from XTAL_PAD when CGPLL 300M generated clock is not available. X38[9] is used to select the switching mechanism. X38[9] =0, HW dynamic switching between 300M and 48M according whether CGPLL is available or not. X38[9]=1, FW switch thru x38[10] 0: 300MHz generated by CGPLL 1: 48MHz generated from XTAL_PAD
9	<b>usb2_host_clk_switching_mechanism.</b> Read-write. Reset: 0. <b>Description:</b> USB2 Host Clock Switching Mechanism USB2 Host Clock can be either 300M generated from CGPLL or 48M generated from XTAL_PAD when CGPLL 300M generated clock is not available. X38[9] is used to select the switching mechanism. 0: HW dynamic switching between 300M and 48M according whether CGPLL is available or not. 1: FW switch thru x38[10]
8	Reserved.
7	<b>usb_60m_clk_staticoff.</b> Read-write. Reset: 0. <b>Description:</b> USB 60M Clock (for HC2) Static Off If USB HC2 is not used in a given platform, USB_60M_Clk can be turn off by set this bit to "1". 0: Static Off function is not enabled 1: Static Off function is enabled
6	<b>usb1_host_clk_staticoff.</b> Read-write. Reset: 0. <b>Description:</b> USB1 Host Clock Static Off If USB1 is not used in a given platform, USB1_Host_Clk can be turn off by set this bit to "1". 0: Static Off function is not enabled 1: Static Off function is enabled
5	<b>usb0_host_clk_staticoff.</b> Read-write. Reset: 0. <b>Description:</b> USB0 Host Clock Static Off If USB0 is not used in a given platform, USB0_Host_Clk can be turn off by set this bit to "1". 0: Static Off function is not enabled 1: Static Off function is enabled
4:0	Reserved.

#### MISCx0000003C (FCH::MISC::cgpll\_configuration3)

Read-write. Reset: 0000\_1000h.

\_aliasHOST; MISCx0000003C; MISC=FED8\_0E00h

Bits	Description
31:13	Reserved.
12	<b>xtal_zstate_refclk_s0_clken.</b> Read-write. Reset: 1. <b>Description:</b> XTAL_PAD Output XTAL_ZSTATE_REFCLK_S0 Enable/disable 0: Turn off 48MHz for XTAL_ZSTATE_REFCLK_S0 by ZSC when in Z state. 1: Turn on, free running 48MHz
11:0	Reserved.

**MISCx00000040 (FCH::MISC::miscclkcntrl0)**

Read-write. Reset: 0000\_0004h.

\_aliasHOST; MISCx00000040; MISC=FED8\_0E00h

Bits	Description
31	Reserved.
30	<b>cg1_cfg_update_req.</b> Read-write. Reset: 0. <b>Description:</b> Set this bit will request CG1_PLL to load spread related value into CG1_PLL. The bit will be clear "0" by hardware after request send to CG1_PLL.
29:26	Reserved.
25	<b>cg1_fbdiv_loaden.</b> Read-write. Reset: 0. Set "1" to enable loading CG1_PLL FB_DIV value form register
24:5	Reserved.
4	<b>bp_x48m0_s0i3_dis.</b> Read-write. Reset: 0. <b>Description:</b> BP_X48M0 Clock Output Disable when at S0A3 This is S0 type clock and will be OFF in sleep state. When this bit is set, 48M clock will be OFF when at S0A3.
3	<b>bp_x48m_s0i2_1_dis.</b> Read-write. Reset: 0. <b>Description:</b> GPIO_48M (24M) Clock Turn Off at S0i2.1 State 0: GPIO_48M will not be turn off at S0i2.1 state 1: GPIO_48M will be turn off at S0i2.1 state
2	<b>bp_x48m0_output_enable.</b> Read-write. Reset: 1. <b>Description:</b> BP_X48M0 Clock Output Enable This is S0 type clock and will be OFF in sleep state. If this GPIO is unused, program this bit to "0" to turn off output for power saving..
1:0	Reserved.

**MISCx00000044 (FCH::MISC::miscclkcntrl1)**

Reset: 0000\_0000h.

\_aliasHOST; MISCx00000044; MISC=FED8\_0E00h

Bits	Description
31:0	Reserved.

**MISCx00000048 (FCH::MISC::miscclkcntl2)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx00000048; MISC=FED8\_0E00h

Bits	Description
31:17	Reserved.
16	<b>mp2_az_48mclk_source_sel.</b> Read-write. Reset: 0. <b>Description:</b> MP2_AZ_48MCLK Source Selection 48M clock is generated by XTAL_PAD. When it turn off and turn-back on again, the 48M output will be oscilated and then stabilitized. 0: Select raw 48M clock 1: Select 48M clock from gater which is enable when 48M stabilitized.
15:12	Reserved.
11	<b>az_48m_clk_s0i21_dis.</b> Read-write. Reset: 0. <b>Description:</b> AZ_48M Clock Turn Off at S0i21 State 0: AZ_48M Clock will not be turn off at S0i21 state 1: AZ_48M Clock will be turn off at S0i21 state
10	<b>az_48m_clk_static_off.</b> Read-write. Reset: 0. <b>Description:</b> AZ_48M Clock Static Turn Off 0: AZ_48M Clock will not be static turn off 1: AZ_48M Clock will be static turn off
9:6	Reserved.
5	<b>mp2_48m_clk_static_off.</b> Read-write. Reset: 0. <b>Description:</b> MP2_48M Clock Static Turn Off 0: MP2_48M Clock will not be static turn off 1: MP2_48M Clock will be static turn off
4:3	Reserved.
2	<b>acp96m_clk_static_off.</b> Read-write. Reset: 0. <b>Description:</b> ACP96M Clock Static Turn Off 0: ACP96M Clock will not be static turn off 1: ACP96M Clock will be static turn off
1:0	Reserved.

**MISCx0000004C (FCH::MISC::miscclkcntl3)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx0000004C; MISC=FED8\_0E00h

Bits	Description
31:12	Reserved.
11	<b>pcie_lane_0_to_19_phy_refclk_static_pwdn_cntl.</b> Read-write. Reset: 0. <b>Description:</b> PCIE_LANE_0_TO_19_PHY Refclk Driver Static PWDN Control 0: Enable on-chip driver (default, static enable) 1: Disable on-chip driver (if it is unused) When resume, it will be static shut-down with on-chip driver enable (default setting) regardless what has been programmed and BIOS needs to program it again.
10	<b>pcie_lane_0_to_19_phy_refclk_pwdn_sel.</b> Read-write. Reset: 0. <b>Description:</b> PCIE_LANE_0_TO_19_PHY Refclk Driver PWDN Selection 0: Static PWDN control by x4C[11] (default) 1: Dynamic PWDN when it is idle/L1SS (dynamic control by DXIO block) When resume, it will be static shut-down with on-chip driver enable (default setting) regardless what has been programmed and BIOS needs to program it again.
9	Reserved.
8	<b>usb7_host_clk_enb.</b> Read-write. Reset: 0. <b>Description:</b> USB7 Host Clock EnableB/Disable Non Sticky bit. When resume, it will be enable (default setting "0") regardless what has been programmed and BIOS needs to program it again if this bit has been programmed to "1". 0: Enable 1: Disable
7	<b>usb3_host_clk_enb.</b> Read-write. Reset: 0. <b>Description:</b> USB3 Host Clock EnableB/Disable Non Sticky bit. When resume, it will be enable (default setting "0") regardless what has been programmed and BIOS needs to program it again if this bit has been programmed to "1". 0: Enable 1: Disable
6:5	Reserved.
4	<b>sata0_sgpio_clk_enb.</b> Read-write. Reset: 0. <b>Description:</b> SATA0 Core Clock (32KHz) EnableB/Disable FCH_SATA0_SGPIO_CLK is generated from FCH. Non Sticky bit. When resume, it will be enable (default setting "0") regardless what has been programmed and BIOS needs to program it again if this bit has been programmed to "1". 0: Enable 1: Disable
3	<b>sata0_core_clk_enb.</b> Read-write. Reset: 0. <b>Description:</b> SATA0 Core Clock (100M) EnableB/Disable FCH_SATA0_100M_CLK which is generated from FCH. Non Sticky bit. When resume, it will be enable (default setting "0") regardless what has been programmed and BIOS needs to program it again if this bit has been programmed to "1". 0: Enable 1: Disable
2	<b>usb1_host_clk_enb.</b> Read-write. Reset: 0.

	<b>Description:</b> USB1 Host Clock EnableB/Disable Non Sticky bit. When resume, it will be enable (default setting "0") regardless what has been programmed and BIOS needs to program it again if this bit has been programmed to "1". 0: Enable 1: Disable
1	<b>usb0_host_clk_enb.</b> Read-write. Reset: 0. <b>Description:</b> USB0 Host Clock EnableB/Disable Non Sticky bit. When resume, it will be enable (default setting "0") regardless what has been programmed and BIOS needs to program it again if this bit has been programmed to "1". 0: Enable 1: Disable
0	<b>usb_hc2_clk_enb.</b> Read-write. Reset: 0. <b>Description:</b> USB HC2 Clock EnableB/Disable Non Sticky bit. When resume, it will be enable (default setting "0") regardless what has been programmed and BIOS needs to program it again if this bit has been programmed to "1". 0: Enable 1: Disable

**MISCx00000050 (FCH::MISC::jtagcntrl)**

Read-write. Reset: 00h.

\_aliasHOST; MISCx00000050; MISC=FED8\_0E00h

Bits	Description
7:0	Reserved.

**MISCx00000058 (FCH::MISC::clkcntrl58)**

Reset: 0000\_0000h.

\_aliasHOST; MISCx00000058; MISC=FED8\_0E00h

Bits	Description
31:0	Reserved.

**MISCx0000005C (FCH::MISC::clkcntrl5c)**

Reset: 0000\_0000h.

\_aliasHOST; MISCx0000005C; MISC=FED8\_0E00h

Bits	Description
31:0	Reserved.

**MISCx00000060 (FCH::MISC::idlecntrl)**

Reset: 0000\_0000h.

\_aliasHOST; MISCx00000060; MISC=FED8\_0E00h

Bits	Description
31:0	Reserved.



**MISCx00000064 (FCH::MISC::dsmxtrig\_routing)**

Read-write. Reset: FFFFh.

\_aliasHOST; MISCx00000064; MISC=FED8\_0E00h

Bits	Description
15	<b>dsmxtrig3_to_ec.</b> Read-write. Reset: 1. <b>Description:</b> 1: Enable CNB2ACPI_AL_trigB[3] low-pulse will trigger a 8051 (EC) action 0: Disabled
14	<b>dsmxtrig2_to_ec.</b> Read-write. Reset: 1. <b>Description:</b> 1: Enable CNB2ACPI_AL_trigB[2] low-pulse will trigger a 8051 (EC) action 0: Disabled
13	<b>dsmxtrig1_to_ec.</b> Read-write. Reset: 1. <b>Description:</b> 1: Enable CNB2ACPI_AL_trigB[1] low-pulse will trigger a 8051 (EC) action 0: Disabled
12	<b>dsmxtrig0_to_ec.</b> Read-write. Reset: 1. <b>Description:</b> 1: Enable CNB2ACPI_AL_trigB[0] low-pulse will trigger a 8051 (EC) action 0: Disabled
11	<b>dsmxtrig3_to_autotransaction.</b> Read-write. Reset: 1. <b>Description:</b> 1: Enable the falling of CNB2ACPI_AL_trigB[3] to trigger a pre-programmed transaction. 0: Disabled
10	<b>dsmxtrig2_to_autotransaction.</b> Read-write. Reset: 1. <b>Description:</b> 1: Enable the falling of CNB2ACPI_AL_trigB[2] to trigger a pre-programmed transaction. 0: Disabled
9	<b>dsmxtrig1_to_autotransaction.</b> Read-write. Reset: 1. <b>Description:</b> 1: Enable the falling of CNB2ACPI_AL_trigB[1] to trigger a pre-programmed transaction. 0: Disabled
8	<b>dsmxtrig0_to_autotransaction.</b> Read-write. Reset: 1. <b>Description:</b> 1: Enable the falling of CNB2ACPI_AL_trigB[0] to trigger a pre-programmed transaction. 0: Disabled
7	<b>dsmxtrig3_to_irq.</b> Read-write. Reset: 1. <b>Description:</b> 1: Enable CNB2ACPI_AL_trigB[3] being low to trigger a specific IRQ (depend on the setting in IO C00/C01 registers) 0: Disabled
6	<b>dsmxtrig2_to_irq.</b> Read-write. Reset: 1. <b>Description:</b> 1: Enable CNB2ACPI_AL_trigB[2] being low to trigger a specific IRQ (depend on the setting in IO C00/C01 registers) 0: Disabled
5	<b>dsmxtrig1_to_irq.</b> Read-write. Reset: 1. <b>Description:</b> 1: Enable CNB2ACPI_AL_trigB[1] being low to trigger a specific IRQ (depend on the setting in IO C00/C01 registers) 0: Disabled
4	<b>dsmxtrig0_to_irq.</b> Read-write. Reset: 1. <b>Description:</b> 1: Enable CNB2ACPI_AL_trigB[0] being low to trigger a specific IRQ (depend on the setting in IO C00/C01 registers) 0: Disabled
3	<b>dsmxtrig3_to_smi.</b> Read-write. Reset: 1. <b>Description:</b> 1: Enable the falling of CNB2ACPI_AL_trigB[3] to trigger an SMI 0: Disabled
2	<b>dsmxtrig2_to_smi.</b> Read-write. Reset: 1. <b>Description:</b> 1: Enable the falling of CNB2ACPI_AL_trigB[2] to trigger an SMI 0: Disabled

1	<b>dsmxtrig1_to_smi.</b> Read-write. Reset: 1.
	<b>Description:</b> 1: Enable the falling of CNB2ACPI_AL_trigB[1] to trigger an SMI 0: Disabled
0	<b>dsmxtrig0_to_smi.</b> Read-write. Reset: 1.
	<b>Description:</b> 1: Enable the falling of CNB2ACPI_AL_trigB[0] to trigger an SMI 0: Disabled

**MISCx00000068 (FCH::MISC::mempwrsavcntrl)**

Read-write.

\_aliasHOST; MISCx00000068; MISC=FED8\_0E00h

Bits	Description
31	<b>ec_bypass_mem_dsd.</b> Read-write. Reset: 1. EC memory BypassMemdsd control.
30	<b>bios_ram_bypass_mem_dsd.</b> Read-write. Reset: 1. <b>Description:</b> BIOS RAM BypassMemdsd control. 0: Enable memory DS features. 1: Disable memory DS. Default is 1'b1, memory DS is disabled.
29	<b>shdw_ram_bypass_mem_dsd.</b> Read-write. Reset: 1. <b>Description:</b> Shadow RAM BypassMemdsd control. 0: Enable memory DS features. 1: Disable memory DS. Default is 1'b1, memory DS is disabled.
28	<b>hub_mem_slp_dis.</b> Read-write. Reset: 0. USB Hub BypassMemdsd control
27	<b>ehci_mem_slp_dis.</b> Read-write. Reset: 0. USB2 BypassMemdsd control
26:21	Reserved.
20	<b>gbe_bypass_mem_dsd.</b> Read-write. Reset: 1. <b>Description:</b> GBE Memory BypassMemdsd control 1: Disable Memory Shutdown. 0: Enable Memory Shutdown. Default is 1'b1, memory Shutdown is disabled.
19	<b>ufs_bypass_mem_dsd.</b> Read-write. Reset: 1. <b>Description:</b> UFS Memory BypassMemdsd control 1: Disable Memory shutdown. 0: Enable Memory shutdown. Default is 1'b1, memory Shutdown is disabled. Note: While PCI reset assert, memory will keep shutdown regardless this bit.
18	<b>sd_bypass_mem_dsd.</b> Read-write. Reset: 1. SD memory BypassMemdsd control.
17	<b>sata_bypass_mem_dsd.</b> Read-write. Reset: 0. SATAMemory BypassMemdsd control.
16	<b>cfga_xhc_ncpu_mem_slp_dis.</b> Read-write. Reset: 0. USB3 memory BypassMemdsd control.
15	<b>hid_bypass_mem_dsd.</b> Read-write. Reset: 1. <b>Description:</b> HID memory (2 of them, 1024x32 and 512x32) BypassMemdsd control. 0: Enable memory DS and SD features. 1: Disable memory DS and SD. Default is 1'b1, memory DS/SD is disabled.
14:6	Reserved.
5	<b>i3c_bypass_mem_dsd.</b> Read-write. Reset: 1. I3C memory BypassMemdsd control.
4	<b>ila_bypass_mem_dsd.</b> Read-write. Reset: 1. ILA memory BypassMemdsd control.
3	<b>amba_bypass_mem_dsd.</b> Read-write. Reset: 1. <b>Description:</b> AMBA memory BypassMemdsd control. 0: Enable memory DS and SD features. 1: Disable memory DS and SD. Default is 1'b1, memory DS/SD is disabled.
2	<b>ab_bypass_mem_dsd.</b> Read-write. Reset: 1. AB memory BypassMemdsd control.
1	<b>sbg_bypass_mem_dsd.</b> Read-write. Reset: 1.

	<b>Description:</b> SBG memory BypassMemdsd control. 0: Enable memory DS and SD features. 1: Disable memory DS and SD. Default is 1'b1, memory DS/SD is disabled.
0	Reserved.

**MISCx00000070 (FCH::MISC::oscfreqcounter)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx00000070; MISC=FED8\_0E00h

Bits	Description
31	<b>countenable.</b> Read-write. Reset: 0. When set, it enables the internal counter to count the number of OSC clocks. When SW is not using this function, it should always set it back to 0 to conserve power.
30	<b>countisvalid.</b> Read-write. Reset: 0. When OscCountPerSec is valid, this bit is set. This bit is read only. SW should always wait for this bit to be set before it should read OscCountPerSec
29:28	Reserved.
27:0	<b>osccountpersec.</b> Read-write. Reset: 000_0000h. Number of OSC clocks per 1 second. Whenever bit 31 (CountEnable) is set, an internal counter will start counting the number of OSC clocks per second and record the count value here.

**MISCx00000074 (FCH::MISC::hpetclkperiod)**

Read-write. Reset: 0429\_B17Eh.

\_aliasHOST; MISCx00000074; MISC=FED8\_0E00h

Bits	Description
31:0	<b>hpetclkperiod.</b> Read-write. Reset: 0429_B17Eh. The register controls the value of clkperiod register in HPET MMIO register space.

**MISCx00000078 (FCH::MISC::postcode)**

Read-only. Reset: 0000\_0000h.

\_aliasHOST; MISCx00000078; MISC=FED8\_0E00h

Bits	Description
31:0	<b>postcode_31_0.</b> Read-only. Reset: 0000_0000h. <b>Description:</b> IO-Wr 83h~80h can write an internal 32-bit PostCode Register (PostCode[31:0]). IO-Rd 80h will return PostCode[7:0]. Reading Misc_Reg:7Bh~78h with PostCodeWidthSel=1 will return {24'b0, PostCode[7:0]}. Reading Misc_Reg:7Bh~78h with PostCodeWidthSel=0 will return PostCode[31:0].

**MISCx0000007C (FCH::MISC::postcodestack)**

Read-only. Reset: FFFF\_FFFFh.

\_aliasHOST; MISCx0000007C; MISC=FED8\_0E00h

Bits	Description
31:0	<b>postcodestack_31_0.</b> Read-only. Reset: FFFF_FFFFh. <b>Description:</b> 8 deep post code STACK read out window. Each time read will get 32 bits post code. Unused byte will return all 0s. If write full, will lost the oldest data and fill in the new data. When SW read out the STACK, it will read from new data to old data, and don't flush them. Extra read will get duplicate data.

**MISCx00000080 (FCH::MISC::strapstatus)**

Read-only.

\_aliasHOST; MISCx00000080; MISC=FED8\_0E00h

Bits	Description
31:29	<b>packageid.</b> Read-only. Reset: 0h. Specify the package ID of the current Die.
28	<b>dieidbkg_id.</b> Read-only. Reset: 0. Specify the Die ID of the current Die.Tie off
27:18	Reserved.
17	<b>clkgenstrap.</b> Read-only. Reset: X. <b>Description:</b> 1: Select 48MHz Crystal clock 0: Select 100MHz PCIe® clock
16:12	Reserved.
11	<b>shortresetstrap.</b> Read-only. Reset: 0. <b>Description:</b> 1: generate short reset, used in simulation 0: Normal reset timing.
10	<b>pcipllbystrap.</b> Read-only. Reset: 0. <b>Description:</b> strap input is ATECONFIG[17] 1: Bypass PCI PLL (used in functional test at tester) 0: not Bypass PCI PLL (Normal operation)
9:6	Reserved.
5:4	<b>romtype.</b> Read-only. Reset: 0h. <b>Description:</b> Boot Rom Type selection: 00: SPI ROM 01: Reserved 10: eSPI with SAFS support 11: eSPI without SAFS support
3:0	Reserved.

**MISCx00000084 (FCH::MISC::hsp\_security)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx00000084; MISC=FED8\_0E00h

Bits	Description
31	<b>hspsecviostatus.</b> Read-write. Reset: 0.
30:17	Reserved.
16	<b>hspsecviorstdisable.</b> Read-write. Reset: 0. Disable cold reset upon security violation detection.
15:1	Reserved.
0	<b>hspsecviodetenable.</b> Read-write. Reset: 0. When enabled, security violation status will be logged in HspSecVioStatus register, and a cold reset will be generated if HspSecVioRstDisable is not set.

**MISCx00000088 (FCH::MISC::postcode\_control)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx00000088; MISC=FED8\_0E00h

Bits	Description
31:1	Reserved.
0	<b>post_rd_ptr_rst.</b> Read-write. Reset: 0. Software reset bit for read pointer. When new turn of read needed, set this bit and read pointer will go back for next turn's read.

**MISCx0000008C (FCH::MISC::dhcp\_lease\_timer)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx0000008C; MISC=FED8\_0E00h

Bits	Description
31:0	<b>dhcpleasetimer.</b> Read-write. Reset: 0000_0000h. <b>Description:</b> Writing this register will update DHCP Lease Timer initial value. If it's non-zero, timer will start counting and a timer expiration event will be generated when it counts down to 0. Reading this register will get current DHCP Lease Timer value.

**MISCx00000090 (FCH::MISC::autotransaction)**

Read-write. Reset: 00h.

\_aliasHOST; MISCx00000090; MISC=FED8\_0E00h

Bits	Description
7:4	<b>transactiontype.</b> Read-write. Reset: 0h. PCI Command type used for this transaction. For example, if this is programmed with a value of 3h, the transaction issue will be an IO write.
3:2	<b>bytecount.</b> Read-write. Reset: 0h. <b>Description:</b> 00: 1 byte 01: 2 bytes 10: 4 bytes 11: 4 bytes
1	<b>dualaddr.</b> Read-write. Reset: 0. <b>Description:</b> 0: Use single address cycle 1: Use dual address cycle
0	<b>autoexecute.</b> Read-write. Reset: 0. Writing this bit will cause the HW to execute the transaction defined by the definition below. Once it is written, this bit stays as 1 until the transaction is completed, in which case it will return to 0

**MISCx00000091 (FCH::MISC::allowec)**

Read-write. Reset: 00h.

\_aliasHOST; MISCx00000091; MISC=FED8\_0E00h

Bits	Description
7:3	Reserved.
2	<b>autotriggerfromcpuen.</b> Read-write. Reset: 0. If this bit is set, a falling edge on KSO15/XDB[1] /(&cnb_fch_dsm_xtrig[3:0]) will trigger this autotransaction logic
1	<b>disableauto.</b> Read-write. Reset: 0. If this bit is set, the entire AutoTransaction logic is disabled. Once this bit is set, it cannot be cleared except by system reset
0	<b>allowectoautotransacten.</b> Read-write. Reset: 0. When this bit is 0, EC cannot write to any of registers relating to any of the registers in the Auto Transaction Generation logic. When this bit is 1, then EC can change any of these bits. Only BIOS can change this bit.

**MISCx00000094 (FCH::MISC::autoaddrlow)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx00000094; MISC=FED8\_0E00h

Bits	Description
31:0	<b>autoaddrlow.</b> Read-write. Reset: 0000_0000h. Low address to be used by the AutoExecute operation

**MISCx00000098 (FCH::MISC::autoaddrhigh)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx00000098; MISC=FED8\_0E00h

Bits	Description
31:0	<b>autoaddrhigh.</b> Read-write. Reset: 0000_0000h. High address to be used by the AutoExecute operation. This register is only applicable when DualAddr = 1.

**MISCx0000009C (FCH::MISC::autodata)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx0000009C; MISC=FED8\_0E00h

Bits	Description
31:0	<b>autodata.</b> Read-write. Reset: 0000_0000h. If the operation is read, this register will return the read data. If the TransactionType is a write command, this register will contain the write data. Note byte is aligned accordingly.

**MISCx000000A0 (FCH::MISC::cgpllcntrl0)**

Reset: 0000\_0000h.

\_aliasHOST; MISCx000000A0; MISC=FED8\_0E00h

Bits	Description
31:0	Reserved.

**MISCx000000A4 (FCH::MISC::cgpllcntrl1)**

Reset: 0000\_0000h.

\_aliasHOST; MISCx000000A4; MISC=FED8\_0E00h

Bits	Description
31:0	Reserved.

**MISCx000000A8 (FCH::MISC::cgpllcntrl2)**

Reset: 0000\_0000h.

\_aliasHOST; MISCx000000A8; MISC=FED8\_0E00h

Bits	Description
31:0	Reserved.

<b>MISCx000000AC (FCH::MISC::cgpllcntrl3)</b>	
Read-write. Reset: 0000_0000h.	
_aliasHOST; MISCx000000AC; MISC=FED8_0E00h	
Bits	Description
31:28	Reserved.
27:26	<b>gpp6_refclk_selection_override.</b> Read-write. Reset: 0h.
	<b>Description:</b> GPP6 Refclk selection Override Program these two bits will override the following selection default. 00 = CG_PLL generated 100MHz 01 = EXT_GPP0_SRC 10 = 48MHz 11 = Reserved Note: Misc_Reg xBC[13] to select EXT_GPP0_SRC either is from GPP0 external input or GPP0 external input with divided-by-2
25:24	Reserved.
23:22	<b>cpu_refclk_selection_override.</b> Read-write. Reset: 0h.
	<b>Description:</b> CPU Refclk Selection Override (XOR) Program these two bits will override the following selection default. 00 = 100MHz CG1_PLL generated (default) 01 = EXT_GPP0_SRC 10 = External input thru GPP1 (external CPU overclock source in eCLK mode) 11 = Reserved Note: Misc_Reg xBC[13] to select EXT_GPP0_SRC either is from GPP0 external input or GPP0 external input with divided-by-2 Should apply a cold reset with CPU refclk source change
21:20	<b>smu_pll_refclk_selection_override.</b> Read-write. Reset: 0h.
	<b>Description:</b> SMU_PLL Refclk Selection Override (XOR) Program these two bits will override the following selection default. 00 = 100MHz CG1_PLL generated (default) 01 = 32KHz 10 = EXT_GPP0_SRC 11 = Reserved Note: Misc_Reg xBC[13] to select EXT_GPP0_SRC either is from GPP0 external input or GPP0 external input with divided-by-2
19:6	Reserved.
5:4	<b>disp_pll_refclk_refclk_selection.</b> Read-write. Reset: 0h.
	<b>Description:</b> DISP_PLL_REFCLK Refclk Selection Used as the reference clock to the PLL that drives the DCN (display) clocks 00 = 48MHz (iCLK/eCLK mde default) 01 = 32KHz 10 = EXT_GPP0_SRC 11 = Reserved Note: Misc_Reg xBC[13] to select EXT_GPP0_SRC either is from GPP0 external input or GPP0 external input with divided-by-2
3	<b>cpl_dfl_ref_clk_refclk_selection.</b> Read-write. Reset: 0.
	<b>Description:</b> Cpl_DFLL_REF_CLK Refclk Selection Used as the reference clock for the DFLL which drives GFXCLK 0 = 48MHz (iCLK/eCLK mde default) 1 = EXT_GPP0_SRC Note: Misc_Reg xBC[13] to select EXT_GPP0_SRC either is from GPP0 external input or GPP0 external input with divided-by-2



2	Reserved.
1:0	<b>pcie_lane_0_to_19_phy_refclk_selection_override.</b> Read-write. Reset: 0h. <b>Description:</b> PCIE_LANE_0_TO_19_PHY Refclk Selection Override Program xAC[1:0] to override the mux selection shown below from iCLK or eCLK mode default if needed. 00 = 100MHz CG1_PLL generated (iCLK/eCLK mde default) 01 = EXT_GPP0_SRC 10 = 48 MHz, buffered version from CG_XTAL input 11 = Reserved Note: Misc_Reg xBC[13] to select EXT_GPP0_SRC either is from GPP0 external input or GPP0 external input with divided-by-2

#### MISCx000000B0 (FCH::MISC::cgpllcntrl4)

Reset: 0000\_0000h.

\_aliasHOST; MISCx000000B0; MISC=FED8\_0E00h

Bits	Description
31:0	Reserved.

#### MISCx000000B4 (FCH::MISC::cgpllcntrl5)

Reset: 0000\_0000h.

\_aliasHOST; MISCx000000B4; MISC=FED8\_0E00h

Bits	Description
31:0	Reserved.

#### MISCx000000B8 (FCH::MISC::cgpllcntrl6)

Reset: 0000\_0000h.

\_aliasHOST; MISCx000000B8; MISC=FED8\_0E00h

Bits	Description
31:0	Reserved.

<b>MISCx000000BC (FCH::MISC::cgpllcntrl7)</b>	
Read-write. Reset: 0000_0000h.	
_aliasHOST; MISCx000000BC; MISC=FED8_0E00h	
<b>Bits</b>	<b>Description</b>
31:28	Reserved.
27:26	<b>gpp5_refclk_selection_override.</b> Read-write. Reset: 0h.  <b>Description:</b> GPP5 Refclk selection Override Program these two bits will override the following selection default. 00 = CG_PLL generated 100MHz 01 = EXT_GPP0_SRC 10 = 48MHz 11 = Reserved Note: Misc_Reg xBC[13] to select EXT_GPP0_SRC either is from GPP0 external input or GPP0 external input with divided-by-2
25:24	<b>gpp4_refclk_selection_override.</b> Read-write. Reset: 0h.  <b>Description:</b> GPP4 Refclk selection Override Program these two bits will override the following selection default. 00 = CG_PLL generated 100MHz 01 = EXT_GPP0_SRC 10 = 48MHz 11 = Reserved
23:22	<b>gpp3_refclk_selection_override.</b> Read-write. Reset: 0h.  <b>Description:</b> GPP3 Refclk selection Override Program these two bits will override the following selection default. 00 = CG_PLL generated 100MHz 01 = EXT_GPP0_SRC 10 = 48MHz 11 = Reserved Note: Misc_Reg xBC[13] to select EXT_GPP0_SRC either is from GPP0 external input or GPP0 external input with divided-by-2
21:20	<b>gpp2_refclk_selection_override.</b> Read-write. Reset: 0h.  <b>Description:</b> GPP2 Refclk selection Override Program these two bits will override the following selection default. 00 = CG_PLL generated 100MHz 01 = EXT_GPP0_SRC 10 = 48MHz 11 = Reserved Note: Misc_Reg xBC[13] to select EXT_GPP0_SRC either is from GPP0 external input or GPP0 external input with divided-by-2
19:18	<b>gpp1_refclk_selection_override.</b> Read-write. Reset: 0h.  <b>Description:</b> GPP1 Refclk selection Override Program these two bits will override the following selection default. 00 = CG_PLL generated 100MHz 01 = EXT_GPP0_SRC 10 = 48MHz 11 = Reserved Note: Misc_Reg xBC[13] to select EXT_GPP0_SRC either is from GPP0 external input or GPP0 external input with divided-by-2
17:16	<b>gpp0_refclk_selection_override.</b> Read-write. Reset: 0h.

	<b>Description:</b> GPP0 Refclk selection Override Program these two bits will override the following selection default. 00 = CG_PLL generated 100MHz 01 = Reserved 10 = 48MHz 11 = Reserved
15	<b>ext_bypassclk_en.</b> Read-write. Reset: 0. <b>Description:</b> EXT_BYPASSCLK_EN Enable/Disable REF_BYPASSCLK external input thru GFX0 0 = disable 1 = enable
14	<b>clk_cgpll_ext_pwdn.</b> Read-write. Reset: 0. <b>Description:</b> CLK_CGPLL_EXT_PWDN Enable/Disable CGPLL external refclk source thru GPP0 input. 0 = enable 1 = disable
13	<b>ext_gpp0_refclk_sel.</b> Read-write. Reset: 0. <b>Description:</b> EXT_GPP0_REFCLK_SEL 0 = GPP0 external input 1 = GPP0 external input divided-by-2
12	Reserved.
11	<b>cpu_refclk_driver_static_pwdn.</b> Read-write. Reset: 0. <b>Description:</b> CPU Refclk Driver Static PWDN Program this bit will static turn off or on CPU refclk. 0 = enable (turn on) 1 = disable (turn off)
10	<b>smu_pll_refclk_driver_static_en.</b> Read-write. Reset: 0. <b>Description:</b> SMU_PLL Refclk Driver Static EN Program this bit will static turn off or on SMU_PLL refclk. 0 = Turn off 1 = Turn on
9	Reserved.
8	<b>disp_pll_refclk_driver_cntl.</b> Read-write. Reset: 0. <b>Description:</b> DISP_PLL Refclk Driver Cntl Program this bit to "1" allows ZSC_Z10_SROFF signal to turn on/off DISP_PLL refclk driver. 0 = Turn on (default) 1 = Turn on/off driver by ZSC_Z10_SROFF from ZSC
7:0	Reserved.

#### MISCx000000C0 (FCH::MISC::iotrapping0)

Read-write. Reset: 0000h.

\_aliasHOST; MISCx000000C0; MISC=FED8\_0E00h

Bits	Description
15:0	<b>iotrappingadr0.</b> Read-write. Reset: 0000h. Specify the I/O address 0 which causes SMI event.

#### MISCx000000C2 (FCH::MISC::iotrapping1)

Read-write. Reset: 0000h.

\_aliasHOST; MISCx000000C2; MISC=FED8\_0E00h

Bits	Description
15:0	<b>iotrappingadr1.</b> Read-write. Reset: 0000h. Specify the I/O address 1 which causes SMI event.

**MISCx000000C4 (FCH::MISC::iotrapping2)**

Read-write. Reset: 0000h.

\_aliasHOST; MISCx000000C4; MISC=FED8\_0E00h

Bits	Description
15:0	<b>iotrappingadr2</b> . Read-write. Reset: 0000h. Specify the I/O address 2 which causes SMI event.

**MISCx000000C6 (FCH::MISC::iotrapping3)**

Read-write. Reset: 0000h.

\_aliasHOST; MISCx000000C6; MISC=FED8\_0E00h

Bits	Description
15:0	<b>iotrappingadr3</b> . Read-write. Reset: 0000h. Specify the I/O address 3 which causes SMI event.

**MISCx000000C8 (FCH::MISC::cfgtrapping0)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx000000C8; MISC=FED8\_0E00h

Bits	Description
31:2	<b>cfgtrappingadr0</b> . Read-write. Reset: 0000_0000h. Specify the CFG address 0 which causes SMI event.
1:0	Reserved.

**MISCCx000000CC (FCH::MISC::smitrappingrwrdovr)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCCx000000CC; MISC=FED8\_0E00h

Bits	Description
31:29	Reserved.
28	<b>sfh_i3c_sda_status.</b> Read-write. Reset: 0. <b>Description:</b> Ready-Only. Pad input status of SFH I3C SDA.
27	<b>sfh_i3c_scl_status.</b> Read-write. Reset: 0. <b>Description:</b> Ready-Only. Pad input status of SFH I3C SCL.
26	<b>sfh_i3c_override_1.</b> Read-write. Reset: 0. <b>Description:</b> 0x0: SFH I3C SDA is used for MP2 I3C. 0x1: SFH I3C SDA is used for MP2 GPIO 1.
25	<b>sfh_i3c_override_0.</b> Read-write. Reset: 0. <b>Description:</b> 0x0: SFH I3C SCL is used for MP2 I3C. 0x1: SFH I3C SCL is used for MP2 GPIO 0.
24	<b>i2c1sdamodeppod.</b> Read-write. Reset: 0. <b>Description:</b> This bit only valid when the Pad I2C1 SDA configured as normal GPIO function 1: I3C Push-pull Mode 0: I3C Open-drain Mode
23	<b>i2c1sclmodeppod.</b> Read-write. Reset: 0. <b>Description:</b> This bit only valid when the Pad I2C1 SCL configured as normal GPIO function 1: I3C Push-pull Mode 0: I3C Open-drain Mode
22	<b>i2c0sdamodeppod.</b> Read-write. Reset: 0. <b>Description:</b> This bit only valid when the Pad I2C0 SDA configured as normal GPIO function 1: I3C Push-pull Mode 0: I3C Open-drain Mode
21	<b>i2c0sclmodeppod.</b> Read-write. Reset: 0. <b>Description:</b> This bit only valid when the Pad I2C0 SCL configured as normal GPIO function 1: I3C Push-pull Mode 0: I3C Open-drain Mode
20	<b>smbus1sdamodeppod.</b> Read-write. Reset: 0. <b>Description:</b> This bit only valid when the Pad SMBUS1 SDA configured as normal GPIO function 1: I3C Push-pull Mode 0: I3C Open-drain Mode
19	<b>smbus1sclmodeppod.</b> Read-write. Reset: 0. <b>Description:</b> This bit only valid when the Pad SMBUS1 SCL configured as normal GPIO function 1: I3C Push-pull Mode 0: I3C Open-drain Mode
18	<b>smbus0sdamodeppod.</b> Read-write. Reset: 0. <b>Description:</b> This bit only valid when the Pad SMBUS0 SDA configured as normal GPIO function 1: I3C Push-pull Mode 0: I3C Open-drain Mode
17	<b>smbus0sclmodeppod.</b> Read-write. Reset: 0. <b>Description:</b> This bit only valid when the Pad SMBUS0 SCL configured as normal GPIO function 1: I3C Push-pull Mode 0: I3C Open-drain Mode
16	<b>cfgtrappingrw0.</b> Read-write. Reset: 0.

	<b>Description:</b> 0: Trap on CFG read access on the address specified in CfgTrappingAdr0 1: Trap on CFG write access on the address specified in CfgTrappingAdr0
15:13	Reserved.
12	<b>memtrappingrdovr0.</b> Read-write. Reset: 0. Set to 1 to force read data to be replaced by MemRdOvrData0.
11:9	Reserved.
8	<b>memtrappingrw0.</b> Read-write. Reset: 0. <b>Description:</b> 0: Trap on MEM read access on the address specified in MemTrappingAdr0 1: Trap on MEM write access on the address specified in MemTrappingAdr0
7:4	Reserved.
3	<b>iotrappingrw3.</b> Read-write. Reset: 0. <b>Description:</b> 0: Trap on I/O read access on the address specified in IoTrappingAdr3 1: Trap on I/O write access on the address specified in IoTrappingAdr3
2	<b>iotrappingrw2.</b> Read-write. Reset: 0. <b>Description:</b> 0: Trap on I/O read access on the address specified in IoTrappingAdr2 1: Trap on I/O write access on the address specified in IoTrappingAdr2
1	<b>iotrappingrw1.</b> Read-write. Reset: 0. <b>Description:</b> 0: Trap on I/O read access on the address specified in IoTrappingAdr1 1: Trap on I/O write access on the address specified in IoTrappingAdr1
0	<b>iotrappingrw0.</b> Read-write. Reset: 0. <b>Description:</b> 0: Trap on I/O read access on the address specified in IoTrappingAdr0 1: Trap on I/O write access on the address specified in IoTrappingAdr0

**MISCx000000D0 (FCH::MISC::memtrapping0)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx000000D0; MISC=FED8\_0E00h

Bits	Description
31:2	<b>memtrappingadr0.</b> Read-write. Reset: 0000_0000h. Specify the 30-bit MEM address 0 which causes SMI even, lowest 2 bits are ignored.
1:0	Reserved.

**MISCx000000D4 (FCH::MISC::memrdovrdata0)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; MISCx000000D4; MISC=FED8\_0E00h

Bits	Description
31:0	<b>memtrappingrddata0.</b> Read-write. Reset: 0000_0000h. The 32 bit data is used as the return data when the memory read trapping is enabled in MemTrapping0. with MemTrappingRdOvr0 = 1

**MISCx000000D8 (FCH::MISC::i2c0\_padctrl)**

Read-write. Reset: 3C00\_0030h.

\_aliasHOST; MISCx000000D8; MISC=FED8\_0E00h

Bits	Description
31:30	<b>spikercsel_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> Enables spike suppression in I2C mode only. Bit 1 for PAD1 and Bit 0 for PAD0. 1= Filter enable (must be used in conjunction with pin Mode_I3c_I2c = 0 only for 1.8V and 1.1V system) 0= Disable
29:28	<b>mode_1p8_1p1_1_0.</b> Read-write. Reset: 3h. <b>Description:</b> Voltage Select pin. Bit 1 for PAD1 and Bit 0 for PAD0. 1 = 1.8V operation 0 = 1.1V operation
27:26	<b>mode_i3c_i2c_1_0.</b> Read-write. Reset: 3h. <b>Description:</b> I3C or I2C mode enable. Bit 1 for PAD1 and Bit 0 for PAD0. 1= I3C mode 0 = I2C mode
25:24	<b>slewp_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> Active Rise Slew Compensation. Bit 1 for PAD1 and Bit 0 for PAD0. 1= Faster rise slew 0= Disable
23:22	<b>resbiasen_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> When the bias circuit is enabled (BiasCrtEn=1), ResBiasEn determines the bias current type. Bit 1 for PAD1 and Bit 0 for PAD0. 1= Constant-gm resistive current 0= Temperature compensated resistive current
21:20	<b>compisel_1_0.</b> Read-write. Reset: 0h. Unused.
19:18	<b>spare_1_0.</b> Read-write. Reset: 0h. Spare pins. Bit 1 for PAD1 and Bit 0 for PAD0.
17:16	<b>biascrten_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> Bias circuit should be ON only in I2C mode. Bit 1 for PAD1 and Bit 0 for PAD0. 1= Enable for I2C Fast/Fast plus mode slew spec 0= Disable for I2C Standard mode
15	<b>rsel1p1.</b> Read-write. Reset: 0. When asserted decreases or increases resistance by 10% for all RC timers. Must be used in conjunction with pin Mode_I3c_I2c = 0 only.
14	<b>rsel0p9.</b> Read-write. Reset: 0. When asserted decreases or increases resistance by 10% for all RC timers. Must be used in conjunction with pin Mode_I3c_I2c = 0 only.
13	<b>csel1p1.</b> Read-write. Reset: 0. When asserted decreases or increases capacitance by 10% for all RC timers. Must be used in conjunction with pin Mode_I3c_I2c = 0 only.
12	<b>csel0p9.</b> Read-write. Reset: 0. When asserted decreases or increases capacitance by 10% for all RC timers. Must be used in conjunction with pin Mode_I3c_I2c = 0 only.
11:10	<b>spikercen_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> Enables spike suppression in I2C mode only. Bit 1 for PAD1 and Bit 0 for PAD0. 1= Filter enable (must be used in conjunction with pin Mode_I3c_I2c = 0 only for 1.8V and 1.1V system) 0= Disable
9:8	<b>fallslewsel_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> I2C Mode select bits. Bit 1 for PAD1 and Bit 0 for PAD0. Must be used when Mode_I3c_I2c=0 pin. 1= I2C Fast Mode (Tx Freq=400KHz), Fast Plus Mode (Tx Freq=1MHz) 0= I2C Standard Mode, Tx Freq=100KHz.
7:6	<b>slewn_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> Active Fall Slew Compensation. Bit 1 for PAD1 and Bit 0 for PAD0. 1= Faster fall slew 0= Disable

5:4	<b>rxsel_1_0.</b> Read-write. Reset: 3h. <b>Description:</b> Select RX of operation. Bit 1 for PAD1 and Bit 0 for PAD0. 1= 1.8V or 1.1V RX (must be used in conjunction with pin Mode_1p8_1p1 = 1/0 for 1.8V and 1.1V system) 0= RX OFF
3:0	<b>od_rp_sw_3_0.</b> Read-write. Reset: 0h. Open-drain pull-up switch. Bit 3 and 2 for PAD1, and Bit 1 and 0 for PAD0.

**MISCx000000DC (FCH::MISC::i2c1\_padctrl)**

Read-write. Reset: 3C00\_0030h.

\_aliasHOST; MISCx000000DC; MISC=FED8\_0E00h

Bits	Description
31:0	<b>i2c1_padctrl_31_0.</b> Read-write. Reset: 3C00_0030h. <b>Description:</b> I2C1 pad control, bit definition is same as Misc_Reg: D8h (I2C0_PadCtrl)

**MISCx000000E0 (FCH::MISC::i2c2\_padctrl)**

Read-write. Reset: 3C00\_0030h.

\_aliasHOST; MISCx000000E0; MISC=FED8\_0E00h

Bits	Description
31:0	<b>i2c2_padctrl_31_0.</b> Read-write. Reset: 3C00_0030h. <b>Description:</b> I2C2 pad control, bit definition is same as Misc_Reg: D8h (I2C0_PadCtrl)

**MISCx000000E4 (FCH::MISC::i2c3\_padctrl)**

Read-write. Reset: 3000\_0030h.

\_aliasHOST; MISCx000000E4; MISC=FED8\_0E00h

Bits	Description
31:0	<b>i2c3_padctrl_31_0.</b> Read-write. Reset: 3000_0030h. <b>Description:</b> I2C3 pad control, bit definition is same as Misc_Reg: D8h (I2C0_PadCtrl)



**MISCx000000E8 (FCH::MISC::i2c4\_padctrl)**

Read-write. Reset: 0300\_000Ch.

\_aliasHOST; MISCx000000E8; MISC=FED8\_0E00h

Bits	Description
31:28	Reserved.
27:26	<b>resbiasen</b> . Read-write. Reset: 0h. <b>Description:</b> When the bias circuit is enabled (BiasCrtEn=1), ResBiasEn determines the bias current type ResBiasEn=0: Temperature compensated resistive current (default) ResBiasEn=1: Constant-gm resistive current
25:24	<b>compsel</b> . Read-write. Reset: 3h. Depending on I2cRxSel compsel will either increase the speed of the I2C RX comparators or it will enable a high Vil one.
23	<b>pden</b> . Read-write. Reset: 0. Pull-down enable for PAD1, 1=Enable 0=Disable
22:21	<b>spare_1_0</b> . Read-write. Reset: 0h. Spare pin
20:19	<b>biascrten_1_0</b> . Read-write. Reset: 0h. Pbias should be on in Fast/Fast+ Mode internally overridden to turn on by FallSlew setting, to save power can be turned off in I2C Standard Mode FallSlew=00.
18	<b>rsl1p1</b> . Read-write. Reset: 0. When asserted increases resistance by 10% for all RC timers
17	<b>rsl0p9</b> . Read-write. Reset: 0. When asserted decreases resistance by 10% for all RC timers
16	<b>csel1p1</b> . Read-write. Reset: 0. When asserted increases capacitance by 10% for all RC timers
15	<b>csel0p9</b> . Read-write. Reset: 0. <b>Description:</b> We rely on some RC components that can have some variation. This is to allow some flexibility incase models are off resulting in our filter too long/short. When asserted decreases capacitance by 10% for all RC timers
14:13	<b>spikercsel</b> . Read-write. Reset: 0h. Select RC constant for I2C spike suppression, 0=50ns, 1=20ns
12:11	<b>spikercen</b> . Read-write. Reset: 0h. Enable Rx spike suppression, default=0
10:9	<b>slewn</b> . Read-write. Reset: 0h. Enable Changing Strength of pre-drive for fall time compensation by 25%. Slewn=1 Enabled, 0=disabled.
8:7	<b>fallslewsel_1_0</b> . Read-write. Reset: 0h. <b>Description:</b> 00= Standard Mode Tx Freq=100Khz) 01= low speed 12ns..120ns Frequency TX<1MHz (FM/FM+) 10= not used 11= not used
6	<b>pden0</b> . Read-write. Reset: 0. Pull-down enable for PAD0, 1=Enable 0=Disable
5:4	<b>i2crxsel_1_0</b> . Read-write. Reset: 0h. <b>Description:</b> i2cRxSel<1:0>=0b01 Schmitt trigger for 3.3V input i2cRxSel<1:0>=0b10 Schmitt trigger for 3.3V input i2cRxSel<1:0>=0b11 Schmitt trigger for 1.8V input i2cRxSel<1:0>=0b00 All receivers off.
3:0	<b>ng_3_0</b> . Read-write. Reset: Ch. N Strength Control.

**MISCx000000EC (FCH::MISC::i2c5\_padctrl)**

Read-write. Reset: 3000\_0030h.

\_aliasHOST; MISCx000000EC; MISC=FED8\_0E00h

Bits	Description
31:0	Reserved.

**MISCx000000F0 (FCH::MISC::clkcntrlf0)**

Read-write. Reset: 2828\_2828h.

\_aliasHOST; MISCx000000F0; MISC=FED8\_0E00h

Bits	Description
31:27	<b>clkreqb_3_t_power_on_value.</b> Read-write. Reset: 05h. <b>Description:</b> CLKREQb_3_T_POWER_ON Value Along with the CLKREQ3_0_T_POWER_ON Scale sets the minimum amount of time (in us) that the Port must wait in L1.2 Exit after sampling CLKREQb_3 asserted before actively driving the interface. T_POWER_ON is calculated by multiplying the value in this field by th value in the T_POWER_ON Scale field.
26	Reserved.
25:24	<b>clkreqb_3_t_power_on_scale.</b> Read-write. Reset: 0h. <b>Description:</b> CLKREQb_3 T_POWER_ON Scale Range of Values: 00: 2us 01: 10us 10: 100us 11: 2us
23:19	<b>clkreqb_2_t_power_on_value.</b> Read-write. Reset: 05h. <b>Description:</b> CLKREQb_2_T_POWER_ON Value Along with the CLKREQb_2_T_POWER_ON Scale sets the minimum amount of time (in us) that the Port must wait in L1.2 Exit after sampling CLKREQb_2 asserted before actively driving the interface. T_POWER_ON is calculated by multiplying the value in this field by th value in the T_POWER_ON Scale field.
18	Reserved.
17:16	<b>clkreqb_2_t_power_on_scale.</b> Read-write. Reset: 0h. <b>Description:</b> CLKREQb_2 T_POWER_ON Scale Range of Values: 00: 2us 01: 10us 10: 100us 11: 2us
15:11	<b>clkreqb_1_t_power_on_value.</b> Read-write. Reset: 05h. <b>Description:</b> CLKREQb_1_T_POWER_ON Value Along with the CLKREQb_1_T_POWER_ON Scale sets the minimum amount of time (in us) that the Port must wait in L1.2 Exit after sampling CLKREQb_1 asserted before actively driving the interface. T_POWER_ON is calculated by multiplying the value in this field by th value in the T_POWER_ON Scale field.
10	Reserved.
9:8	<b>clkreqb_1_t_power_on_scale.</b> Read-write. Reset: 0h. <b>Description:</b> CLKREQb_1 T_POWER_ON Scale Range of Values: 00: 2us 01: 10us 10: 100us 11: 2us
7:3	<b>clkreqb_0_t_power_on_value.</b> Read-write. Reset: 05h. <b>Description:</b> CLKREQb_0_T_POWER_ON Value Along with the CLKREQb_0_T_POWER_ON Scale sets the minimum amount of time (in us) that the Port must wait in L1.2 Exit after sampling CLKREQb_0 asserted before actively driving the interface. T_POWER_ON is calculated by multiplying the value in this field by th value in the T_POWER_ON Scale field.
2	Reserved.
1:0	<b>clkreqb_0_t_power_on_scale.</b> Read-write. Reset: 0h.

	<b>Description:</b> CLKREQb_0 T_POWER_ON Scale Range of Values: 00: 2us 01: 10us 10: 100us 11: 2us
--	---

#### MISCx000000F4 (FCH::MISC::clkcntrlf4)

Read-write. Reset: 0028\_2828h.

\_aliasHOST; MISCx000000F4; MISC=FED8\_0E00h

Bits	Description
31:24	Reserved.
23:19	<b>clkreqb_6_t_power_on_value.</b> Read-write. Reset: 05h. <b>Description:</b> CLKREQb_6_T_POWER_ON Value Along with the CLKREQb_6_T_POWER_ON Scale sets the minimum amount of time (in us) that the Port must wait in L1.2 Exit after sampling CLKREQb_6 asserted before actively driving the interface. T_POWER_ON is calculated by multiplying the value in this field by th value in the T_POWER_ON Scale field.
18	Reserved.
17:16	<b>clkreqb_6_t_power_on_scale.</b> Read-write. Reset: 0h. <b>Description:</b> CLKREQb_6 T_POWER_ON Scale Range of Values: 00: 2us 01: 10us 10: 100us 11: 2us
15:11	<b>clkreqb_5_t_power_on_value.</b> Read-write. Reset: 05h. <b>Description:</b> CLKREQb_5_T_POWER_ON Value Along with the CLKREQb_5_T_POWER_ON Scale sets the minimum amount of time (in us) that the Port must wait in L1.2 Exit after sampling CLKREQb_5 asserted before actively driving the interface. T_POWER_ON is calculated by multiplying the value in this field by th value in the T_POWER_ON Scale field.
10	Reserved.
9:8	<b>clkreqb_5_t_power_on_scale.</b> Read-write. Reset: 0h. <b>Description:</b> CLKREQb_5 T_POWER_ON Scale Range of Values: 00: 2us 01: 10us 10: 100us 11: 2us
7:3	<b>clkreqb_4_t_power_on_value.</b> Read-write. Reset: 05h. <b>Description:</b> CLKREQb_4_T_POWER_ON Value Along with the CLKREQb_4_T_POWER_ON Scale sets the minimum amount of time (in us) that the Port must wait in L1.2 Exit after sampling CLKREQb_4 asserted before actively driving the interface. T_POWER_ON is calculated by multiplying the value in this field by th value in the T_POWER_ON Scale field.
2	Reserved.
1:0	<b>clkreqb_4_t_power_on_scale.</b> Read-write. Reset: 0h. <b>Description:</b> CLKREQb_4 T_POWER_ON Scale Range of Values: 00: 2us 01: 10us 10: 100us 11: 2us

**MISCx000000F8 (FCH::MISC::tdr\_security\_status)**

Read-only.

\_aliasHOST; MISCx000000F8; MISC=FED8\_0E00h

Bits	Description
31:22	Reserved.
21	<b>tdr_i3c_control_updated.</b> Read-only. Reset: X. Tdr_I3C_CONTROL_accessed
20	<b>tdr_ioconfig_updated.</b> Read-only. Reset: X. Tdr_S5IODIRCONFIG_accessed Tdr_S0IODIRCONFIG_accessed Tdr_S5IODIRAPPEND_accessed Tdr_S0IODIRAPPEND_accessed
19	<b>tdr_s5_clkconfig_updated.</b> Read-only. Reset: X. Tdr_S5_PLLDFTCONFIG_accessed Tdr_S5_PLLDFTSTS_accessed Tdr_S5_CLKDFTCONFIG_accessed
18	<b>tdr_sdio_control_updated.</b> Read-only. Reset: X. Tdr_SDIO_CONTROL_accessed
17	<b>tdr_i2c_control_updated.</b> Read-only. Reset: X. Tdr_I2C_CONTROL_accessed
16	<b>tdr_gpio_control_updated.</b> Read-only. Reset: X. Tdr_GPIO18_CONTROL_accessed Tdr_GPIO33_CONTROL_accessed
15	<b>tdr_iodftcfg_updated.</b> Read-only. Reset: X. Tdr_IODFTCFG_accessed
14	<b>tdr_spare_updated.</b> Read-only. Reset: X. Tdr_SPARE_accessed
13	<b>classb_tdr_updated.</b> Read-only. Reset: X. A TDR that is only enabled after Cpl_aeb_valid has been updated.
12	<b>s0stateobserve_tdr_updated.</b> Read-only. Reset: X. Tdr_S0STATEOBSERVE_accessed
11	<b>ila_tdr_updated.</b> Read-only. Reset: X. Tdr_ILA_accessed
10	<b>idcode_tdr_updated.</b> Read-only. Reset: X. Tdr_IDCODE_accessed   Tdr_PKGID_accessed
9	<b>scan_tdr_updated.</b> Read-only. Reset: X. Tdr_SYSSCAN_accessed   Tdr_SCANCONFIG_accessed
8	<b>sysdebug_tdr_updated.</b> Read-only. Reset: X. Tdr_SYSDEBUG_accessed
7	<b>sdb_tdr_updated.</b> Read-only. Reset: X. Tdr_SDB_CONFIG_accessed   Tdr_SDB_STATUS_accessed
6	<b>mbist_tdr_updated.</b> Read-only. Reset: X. Tdr_MBISTCONFIG_accessed   Tdr_MEMBISTSTS_accessed
5	<b>iotstcntrl_tdr_updated.</b> Read-only. Reset: X. Tdr_IOTSTCNTRL_accessed
4	<b>testclkdis_tdr_updated.</b> Read-only. Reset: X. Tdr_TESTCLKDIS_accessed
3	<b>ateconfig_tdr_updated.</b> Read-only. Reset: X. Tdr_ATECONFIG_accessed
2	<b>pll_clk_dft_tdr_updated.</b> Read-only. Reset: X. Tdr_PLLDFT_accessed   Tdr_CLKDFT_accessed
1	<b>acpi_tdr_updated.</b> Read-only. Reset: X. Tdr_ACPIJTAGCMD_accessed   Tdr_ACPIJTAGSTATUS_accessed
0	<b>any_tdr_updated.</b> Read-only. Reset: X. <b>Description:</b> Set if any of the FCH TDR are Updated (Dfx Security Feature) 0: No FCH TDR has been updated since RsmrstB was asserted 1: One or More FCH TDR registers have been updated

**MISCx000000FC (FCH::MISC::clkgating\_cntrl)**

Read-write. Reset: FFFF\_FFFFh.

\_aliasHOST; MISCx000000FC; MISC=FED8\_0E00h

Bits	Description
31:27	Reserved.
26	<b>hpet64_tmr_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of HPET64 Timer block can be stopped if not being Accessed 1: the clock of HPET64 Timer block can't be stopped.
25	<b>hpet32_tmr_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of HPET32 Timer block can be stopped if not being Accessed 1: the clock of HPET32 Timer block can't be stopped.
24	<b>boot_tmr_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of Boot Timer block can be stopped if not being Accessed 1: the clock of Boot Timer block can't be stopped.
23	<b>long_tmr_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of Long Timer block can be stopped if not being Accessed 1: the clock of Long Timer block can't be stopped.
22	<b>shrt_tmr_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of Short Timer block can be stopped if not being Accessed 1: the clock of Short Timer block can't be stopped.
21	<b>asf_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of ASF block can be stopped if not being Accessed 1: the clock of ASF block can't be stopped.
20	<b>asf_clkgen_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of ASF Clock Generator block can be stopped if not being Accessed 1: the clock of ASF Clock Generator block can't be stopped.
19	<b>shdw_syscnt_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of Shadow System Counter block can be stopped if not being Accessed 1: the clock of Shadow System Counter block can't be stopped.
18	<b>acpi_msi_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of ACPI MSI block can be stopped if not being Accessed. 1: the clock of ACPI MSI block can't be stopped.
17	<b>usb_legacy_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of USB Legacy block can be stopped if not being Accessed. 1: the clock of USB Legacy block can't be stopped.
16	<b>isa_bridge_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of ISA Bridge block can be stopped if not being Accessed. 1: the clock of ISA Bridge block can't be stopped.
15	<b>acpi_debug_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of ACPI debug block can be stopped if not being Accessed. 1: the clock of ACPI debug block can't be stopped.
14	<b>tmr_8254_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of 8254 timer block can be stopped if not being Accessed. 1: the clock of 8254 timer block can't be stopped.
13	<b>shdw_pcislave_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of Shadow pcislave block can be stopped if not being Accessed. 1: the clock of Shadow pcislave block can't be stopped.
12	<b>watchdogtimerblk_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of WatchDogTimer block can be stopped if not being Accessed. 1: the clock of WatchDogTimer block can't be stopped.

11	<b>acdctmrblk_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of AcDcTmr block can be stopped if not being Accessed. 1: the clock of AcDcTmr block can't be stopped.
10	<b>biosramblk_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of Bios Ram block can be stopped if not being Accessed. 1: the clock of Bios Ram block can't be stopped.
9:7	Reserved.
6	<b>obffblk_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of OBFF block can be stopped if not being Accessed. 1: the clock of OBFF block can't be stopped.
5	<b>smbus0_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of Smbus0 block can be stopped if not being Accessed. 1: the clock of Smbus0 block can't be stopped.
4	Reserved.
3	<b>aoacregblk_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of Aoac register block can be stopped if not being Accessed. 1: the clock of Aoac register block can't be stopped.
2	<b>pmio2regblk_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of Pmio2 register block can be stopped if not being Accessed. 1: the clock of Pmio2 register block can't be stopped.
1	<b>pmioregblk_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of Pmio register block can be stopped if not being Accessed. 1: the clock of Pmio register block can't be stopped.
0	<b>miscregblk_clkgatingdis.</b> Read-write. Reset: 1. <b>Description:</b> 0: the clock of MISC register block can be stopped if not being Accessed. 1: the clock of MISC register block can't be stopped.

### 7.3.9.2 Power Management (PM) Registers and Standard ACPI Registers

Table 101: ACPI MMIO Space Allocation

00FFh-0000h	SMBus PCI configuration registers, see .
01FFh-0100h	Reserved.
02FFh-0200h	SMI, see 7.3.3 [SMI Registers].
03FFh-0300h	PMIO and ACPI, see 7.3.9.2 [Power Management (PM) Registers and Standard
08FFh-0800h	ACPI Registers].
05FFh-0500h	BIOS RAM
06FFh-0600h	CMOS RAM
07FFh-0700h	CMOS
09FFh-0900h	ASF registers, see .
0AFFh-0A00h	SMBus registers, see .
0BFFh-0B00h	Watchdog registers, see 7.3.5 [Watchdog Timer (WDT) Registers].
0CFFh-0C00h	HPET, see 7.3.4 [High Precision Event Timer (HPET) Registers].
0DFFh-0D00h	IOMux, see 7.3.10.1 [IOMUX Registers].
0EFFh-0E00h	Miscellaneous registers, see 7.3.9.1 [Miscellaneous (MISC) Registers].
10FFh-1000h	Reserved.
11FFh-1100h	Reserved.
14FFh-1400h	DP-VGA
19FFh-1500h	GPIO Registers, see 7.3.10.2 [GPIO Registers].

1BFFh-1B00h	Reserved.
1CFFh-1C00h	Reserved.
1DFFh-1D00h	Wake Device (AC DC timer), see 7.3.6 [Wake Alarm Device (AcDcTimer) Registers].
1EFFh-1E00h	Reserved.
1FFFh-1F00h	Reserved.

The way of accessing PM registers is through the direct mapping scheme. The direct mapping through Memory Mapped IO is 0xFED8\_03XX.



PMx00000000 (FCH::PM::decodeen)	
Read-write. Reset: E302_0B10h.	
_aliasHOST; PMx00000000; PM=FED8_0300h	
Bits	Description
31:30	<b>ioapicconfig.</b> Read-write. Reset: 3h. Set to 11 to improve IoApic latency
29	<b>hpet_msi_en.</b> Read-write. Reset: 1. Set to 1 to expose MSI cap in HPET Cap register..
28	<b>hpet_width_sel.</b> Read-write. Reset: 0. <b>Description:</b> 0: HPET is 32-bit 1: HPET is 64-bit
27:26	<b>watchdogoptions.</b> Read-write. Reset: 0h. Set to 00 for normal WatchDogTimer operation.
25:24	<b>watchdogfreq.</b> Read-write. Reset: 3h. <b>Description:</b> This registers define the clock frequency used by the WatchDogTimer 00: 32us 01: 10ms 10: 100ms 11: 1s
23:21	<b>asfclkssel.</b> Read-write. Reset: 0h. <b>Description:</b> The value controls the frequency of ASF master clock its definition is: 000: ~100KHz 001: ~200KHz 010: ~300kHz 011:~ 400kHz 100:~ 600kHz 101:~ 800kHz 110:~ 900kHz 111:~1MHz
20:19	<b>smbus0sel.</b> Read-write. Reset: 0h. <b>Description:</b> SmBus port selection. There is only one SMBUS engine controlling four SMBUS ports. This register routes the SMBUS engine to the desired port. 00: Port 0 (For SMBUS on the board) 01: Port 2 (Dedicated for TSI polling)
18	<b>asfclkswitch.</b> Read-write. Reset: 0. Set to 1 to change ASF master clock from RTC(32k) to the clock defined in ASFClkSel of the same register.
17	<b>asfclkstretchen.</b> Read-write. Reset: 1. Set to 1 to enable clock stretching support.
16	<b>asfsmmasteren.</b> Read-write. Reset: 0. Set to 1 to enable ASF SMBUS master function.
15:8	<b>smbusasfiobase.</b> Read-write. Reset: 0Bh. <b>Description:</b> Smbus Io base = {Smbus0AsfIoBase[7:0], 0x00} ASF Io base = {Smbus0AsfIoBase[7:0], 0x20} By default Smbus Io base is 0xB00 and ASF Io base is 0xB20
7	<b>watchdogtmren.</b> Read-write. Reset: 0. Set to 1 to enable WatchDog Timer memory(FEB00000 ~ FEB00003) decoding, and enable WatchDog Timer operation.
6	<b>hpeten.</b> Read-write. Reset: 0. Set to 1 to enable HPET memory(FED00000 ~ FED001FF) decoding.
5	<b>ioapicen.</b> Read-write. Reset: 0. Set to 1 to enable IoApic memory(FEC00000 ~ FEC0007F) decoding.
4	<b>smbusasfioen.</b> Read-write. Reset: 1. Set to 1 to enable Smbus and Asf Io decoding. Smbus and Asf Io range are defined in Smbus0AsfIoBase.
3	<b>dmaport80.</b> Read-write. Reset: 0. Set to 1 to pass Io port 0x80, 0x81, 0x82, 0x83 to legacy Dma Io range.
2	<b>legacydmaioen.</b> Read-write. Reset: 0. Set to 1 to enable legacy Dma Io range.
1	<b>cf9ioen.</b> Read-write. Reset: 0. Set to 1 to enable CF9 Io port decoding
0	<b>legacyioen.</b> Read-write. Reset: 0.



	<b>Description:</b> Set to 1 to enable the following Io decoding: 0x20, 0x21, 0xA0, 0xA1 (PIC) 0x40, 0x41, 0x42, 0x43, 0x61 (8254 timer) 0x70, 0x71, 0x72, 0x73 (Rtc) 0x92
--	--

**PMx00000004 (FCH::PM::isacontrol)**

Read-write. Reset: 0000\_0002h.

PrtcControl registers are reset by RsmRstB and UsrRstB. RsmRstB is asserted during G3 state. UsrRstB is asserted when the system reset button is pressed (in S0 state). dPassEnable will be 0 during G3 state so that when S5 power is down, the invalid values of PrtcControl Registers will be isolated to 0 for PRTC.

One important requirement is:

Rtc32KHz clock's availability should not have any dependency with the PrtcControl Registers. This is because we use Rtc32KHz clock for various reset de-bounce logics including the reset for PrtcControl Registers.

\_aliasHOST; PMx00000004; PM=FED8\_0300h

Bits	Description
31	<b>rtc_bg_adj5.</b> Read-write. Reset: 0. PRTC band-gap control bit
30	<b>rtc_bg_adj4.</b> Read-write. Reset: 0. PRTC band-gap control bit
29	<b>rtc_bg_adj3.</b> Read-write. Reset: 0. PRTC band-gap control bit
28	<b>rtc_bg_adj2.</b> Read-write. Reset: 0. PRTC band-gap control bit
27	<b>rtc_bg_adj1.</b> Read-write. Reset: 0. PRTC band-gap control bit
26	<b>rtc_bg_adj0.</b> Read-write. Reset: 0. PRTC band-gap control bit
25	<b>rtc_bg_set_en.</b> Read-write. Reset: 0. <b>Description:</b> This is the enable for the PRTC 32KHz band-gap control bits (rtc_bg_adj0~5). 0: PRTC band-gap uses its own hard-wired configuration. 1: PRTC band-gap uses the rtc_bg_adj* bits.
24	<b>rtc_high_res.</b> Read-write. Reset: 0. Debug purpose register requested by PRTC designer.
23:22	Reserved.
21	<b>rtc_osc_op1.</b> Read-write. Reset: 0. PRTC oscillator control bit
20	<b>rtc_osc_op0.</b> Read-write. Reset: 0. PRTC oscillator control bit
19	<b>rtc_osc_set1.</b> Read-write. Reset: 0. PRTC oscillator control bit
18	<b>rtc_osc_set0.</b> Read-write. Reset: 0. PRTC oscillator control bit
17	<b>rtc_osc_set_en.</b> Read-write. Reset: 0. This is the latch enable for the PRTC 32KHz oscillator control bits (rtc_osc_set0~1, rtc_osc_op0~1).
16	<b>abclkgateen.</b> Read-write. Reset: 0. <b>Description:</b> Master switch for Alink and Blink clock gating. 0: Disabled 1: Enabled
15	<b>pm_lock_iomux.</b> Read-write. Reset: 0. Set to 1 to lock all IOMUX registers from write, once set to =1, this bit cannot be written 0. This bit is reset by PCIReset.
14:12	Reserved.
11	<b>bm_req_en.</b> Read-write. Reset: 0. Legacy BM_REQ# function enable bit it is now for debug purpose only.
10	Reserved.
9	<b>drqmasken.</b> Read-write. Reset: 0. Set to 1 to allow DMA DRQ input to block clock gating
8	<b>dmaenhanceen.</b> Read-write. Reset: 0. Set to 1 to enable enhancement of legacy Dma.
7:6	Reserved.
5	<b>read_shadow.</b> Read-write. Reset: 0. Set to 1 to allow to read Pic ICWX, OCWX register through Io port 21h/A1h.
4:2	Reserved.
1	<b>mmioen.</b> Read-write. Reset: 1. Set to 1 to enable Acpi Mmio range (FED8_0000 ~FED8_1FFF). The space is allocated as specified in Table shows the Address mapping of each ACPI block in front of register PM_regx00
0	<b>biosramen.</b> Read-write. Reset: 0. Set to 1 to enable bios Ram whose base is FED1_0000 (256 bytes)

**PMx00000008 (FCH::PM::pcicontrol)**

Read-write. Reset: 010C\_0500h.

SLP\_TYP is located in AcpiPm1CntBlk offset 00h, bits 10~12.

\_aliasHOST; PMx00000008; PM=FED8\_0300h

Bits	Description
31:26	Reserved.
25	<b>force_slpstate_retry.</b> Read-write. Reset: 0. Set to 1 to send out SMI message before the completion response of IO write to SLP_TYP register. This is to be used in conjunction with SMI trapping on write to SLP_TYP register
24	<b>force_stpclk_retry.</b> Read-write. Reset: 1. <b>Description:</b> Set to 1 to send out STPCLK message before the completion response of the following 3 types of request: I/O write to Slp_typ register I/O write Ldt_stp command C1e cycle Normally it is required to send out STPCLK before completion of the cycles listed above, except for the case of SMI trapping. In that case, this bit should be left as 0
23	<b>ab_stall_en.</b> Read-write. Reset: 0. Set to 1 to allow the legacy DMA engine to hold the internal bus before completing legacy DMA on the LPC bus. This is only needed for certain old LPC devices.
22:21	Reserved.
20	<b>shutdownoption.</b> Read-write. Reset: 0. <b>Description:</b> 0: Issue Init message upstream when receiving shutdown message. 1 : Generate Pci reset when receiving shutdown message.
19	<b>masternowait.</b> Read-write. Reset: 1. <b>Description:</b> 1: ACPI PCI Master doesn't wait for Slave idle when it wants to request bus. 0: Old behavior. PCI Master will wait for Slave idle. Note: Software need keep this bit to1. In rare case, if there is downstream cycle to ACPI register (from CPU/SMN/M2P) and there is upstream Interrupt message pending, state machine will lock up if this bit is not set to 1.
18	<b>echohostfix.</b> Read-write. Reset: 1. Not used.
17	<b>changedma.</b> Read-write. Reset: 0. Not used.
16	<b>gatedma.</b> Read-write. Reset: 0. Not used.
15	<b>clock_slow_mask.</b> Read-write. Reset: 0. Not used.
14:12	<b>ext_intr_time.</b> Read-write. Reset: 0h. Specify the extended interrupt time in 2 microsecond intervals. This is used for preventing APU from re-entering C state right away when it just breaks out from a C state
11	<b>dly_en.</b> Read-write. Reset: 0. Not used.
10	<b>ignr_usb_smi_req.</b> Read-write. Reset: 1. Not used.
9	<b>block_acpi_s5_intr_st.</b> Read-write. Reset: 0. Set to 1, it will block the acpi_s5 interrupt status to send out.
8	<b>pic_apic_arbiter.</b> Read-write. Reset: 1. Set to 1 to arbitrate between PIC request and IOAPIC request
7	<b>force_smaf_match.</b> Read-write. Reset: 0. Set to 1 to enable STPGNT message matching to the expected SMAF.
6	<b>nmimsgsel.</b> Read-write. Reset: 0. <b>Description:</b> 0: Encode NMI request as legacy PIC NMI message type. 1: NMI request as NMI message type.
5	<b>picmsgsel.</b> Read-write. Reset: 0. <b>Description:</b> 0: Encode PIC interrupt request as Legacy PIC ExtInt message type 1: encode PIC interrupt request as ExtInt message type
4	<b>msg_intr_enable.</b> Read-write. Reset: 0. Set to 1 to deliver legacy PIC interrupt as message type.
3	Reserved.
2	<b>fake_dma_en.</b> Read-write. Reset: 0. Not used
1	<b>undo_wrtd_done.</b> Read-write. Reset: 0. Not used

0	<b>undo_dma_change.</b> Read-write. Reset: 0. Not used
---	--

**PMx0000000C (FCH::PM::stpcclkmaf)**

Read-write. Reset: 5543\_2106h.

\_aliasHOST; PMx0000000C; PM=FED8\_0300h

Bits	Description
31	Reserved.
30:28	<b>ttsmaf.</b> Read-write. Reset: 5h. System management action field for Thermal Throttling STPCLK message
27	Reserved.
26:24	<b>nssmaf.</b> Read-write. Reset: 5h. System management action field for Normal Throttling STPCLK message
23	Reserved.
22:20	<b>s3smaf.</b> Read-write. Reset: 4h. System management action field for S3 STPCLK message
19	Reserved.
18:16	<b>s1smaf.</b> Read-write. Reset: 3h. System management action field for S1 STPCLK message
15	Reserved.
14:12	<b>vfsmaf.</b> Read-write. Reset: 2h. System management action field for VFID STPCLK message
11	Reserved.
10:8	<b>c3smaf.</b> Read-write. Reset: 1h. System management action field for C3 STPCLK message
7	Reserved.
6:4	<b>c2smaf.</b> Read-write. Reset: 0h. System management action field for C2 STPCLK message
3	Reserved.
2:0	<b>s4s5smaf.</b> Read-write. Reset: 6h. System management action field for S4/5 STPCLK message

**PMx00000010 (FCH::PM::pwrrstcfg)**

Read-write.

\_aliasHOST; PMx00000010; PM=FED8\_0300h

Bits	Description
15:7	Reserved.
6	<b>fatalrsttos5rst.</b> Read-write. Reset: 1.
5	<b>rtcclkchken_s5reg.</b> Read-write. Reset: 1. <b>Description:</b> 0: RtcClkChk function is disabled 1: RtcClkChk function depends on the strap pin. If the strapped value is 1, RtcClkChk function is enabled. The strap name is RtcClkChkEn_strap.
4	<b>slps3waitrstasrt.</b> Read-write. Reset: 1. Debug purpose
3:2	<b>rstblkssel_1_0.</b> Read-write. Reset: 1h. <b>Description:</b> 00: Always use old logic for reset signals. 01: Use new logic for reset signals during S0i3 entry/exit, otherwise, use old logic. 10: Always use new logic for reset signals. 11: Always use new logic for reset signals. Note: Old logic: fch_pwr_detect New logic: fch_PwrRst
1	<b>toggleallpwrgoodoncf9.</b> Read-write. Reset: 0. <b>Description:</b> 1: De-assert and then assert all PwrGood signals (for PG1, PG1a, PG2 and XHC) during CF9 reset. 0: During CF9 reset, PG1_PwrGood stay at high. PG1a_PwrGood, PG2_PwrGood and Xhc_PwrGood behavior depend on TogglePG1aPG2PGXHConCf9 register bit during CF9 reset.
0	<b>togglepg1apg2pgxhconcf9.</b> Read-write. Reset: 1. <b>Description:</b> 1: De-assert and then assert PG1a_PwrGood, PG2_PwrGood and Xhc_PwrGood during CF9 reset. 0: PG1a_PwrGood, PG2_PwrGood and Xhc_PwrGood stay at high during CF9 reset.

**PMx00000012 (FCH::PM::fchpwrgoodtmr)**

Read-write.

\_aliasHOST; PMx00000012; PM=FED8\_0300h

Bits	Description
15:11	Reserved.
10:0	<b>fchpwrgoodtmr</b> . Read-write. Reset: 150h. T1, Debouncetimer value for FCHPwrGood, and the delay timer value for PwrRstB deassertion of devices in PG1.

**PMx00000014 (FCH::PM::pgpwrgoodtoallentmr)**

Read-write.

\_aliasHOST; PMx00000014; PM=FED8\_0300h

Bits	Description
15:11	Reserved.
10:0	<b>pgpwrgoodtoallentmr</b> . Read-write. Reset: 016h. T18, Timer for PwrGood to AllEn for all the tiles.

**PMx00000016 (FCH::PM::pgpwrgoodassertiontmr)**

Read-write.

\_aliasHOST; PMx00000016; PM=FED8\_0300h

Bits	Description
15:11	Reserved.
10:0	<b>pgpwrgoodassertiontmr</b> . Read-write. Reset: 020h. T19, Timer for DebouncedPwrGood to PwrGood of all the tiles

**PMx00000018 (FCH::PM::pwrgatetmr)**

Read-write. Reset: A864\_8642h.

\_aliasHOST; PMx00000018; PM=FED8\_0300h

Bits	Description
31:28	<b>pwrislanddaughterassertiontmr</b> . Read-write. Reset: Ah. T24, Timer for PwrOn=0 to Daughter_SD=1 of PG1a
27:24	<b>pwrislandmotherassertiontmr</b> . Read-write. Reset: 8h. T23, Timer for PwrOn=0 to Mother_SD=1 of PG1a
23:20	<b>pwrislandmemdaughterassertiontmr</b> . Read-write. Reset: 6h. T22, Timer for PwrOn=0 to Mem_Daughter_SD=1 of PG1a
19:16	<b>pwrislandmemmotherassertiontmr</b> . Read-write. Reset: 4h. T21, Timer for PwrOn=0 to Mem_Mother_SD=1 of PG1a
15:12	<b>pwrislandmemmotherdeassertiontmr</b> . Read-write. Reset: 8h. T17, Timer for PwrOn=1 to Mem_Mother_SD=0 of PG1a
11:8	<b>pwrislandmemdaughterdeassertiontmr</b> . Read-write. Reset: 6h. T16, Timer for PwrOn=1 to Mem_Daughter_SD=0 of PG1a
7:4	<b>pwrislandmotherdeassertiontmr</b> . Read-write. Reset: 4h. T15, Timer for PwrOn=1 to Mother_SD =0 of PG1a
3:0	<b>pwrislanddaughterdeassertiontmr</b> . Read-write. Reset: 2h. T14, Timer for PwrOn=1 to Daughter_SD =0 of PG1a

**PMx0000001C (FCH::PM::pllrstbtmr)**

Read-write.

\_aliasHOST; PMx0000001C; PM=FED8\_0300h

Bits	Description
15:11	Reserved.
10:0	<b>pllrstbtmr</b> . Read-write. Reset: 040h. Delay timer value for PLLRstB deassertion.

**PMx0000001E (FCH::PM::plllocktmr)**

Read-write.

\_aliasHOST; PMx0000001E; PM=FED8\_0300h

Bits	Description
15:11	Reserved.
10:0	<b>plllocktmr</b> . Read-write. Reset: 03Ch. Delay timer value for PllLock assertion.

**PMx00000020 (FCH::PM::pcirstbtmr)**

Read-write.

\_aliasHOST; PMx00000020; PM=FED8\_0300h

Bits	Description
15:11	Reserved.
10:0	<b>pcirstbtmr</b> . Read-write. Reset: 120h. Delay timer value for PciRstB deassertion.

**PMx00000022 (FCH::PM::cpurstbtmr)**

Read-write.

\_aliasHOST; PMx00000022; PM=FED8\_0300h

Bits	Description
15:11	Reserved.
10:0	<b>cpurstbtmr</b> . Read-write. Reset: 19Ch. Delaytimer value for CpuRstB deassertion.

**PMx00000024 (FCH::PM::nbpwrgoodtmr)**

Read-write.

\_aliasHOST; PMx00000024; PM=FED8\_0300h

Bits	Description
15:11	Reserved.
10:0	<b>nbpwrgoodtmr</b> . Read-write. Reset: 0A0h. Delay timer value for NBPwrGood assertion

**PMx00000026 (FCH::PM::cpupwrgoodtmr)**

Read-write.

\_aliasHOST; PMx00000026; PM=FED8\_0300h

Bits	Description
15:11	Reserved.
10:0	<b>cpupwrgoodtmr</b> . Read-write. Reset: 0E0h. Delay timer value for CpuPwrGood assertion

**PMx00000028 (FCH::PM::s0tos5entmr)**

Read-write.

\_aliasHOST; PMx00000028; PM=FED8\_0300h

Bits	Description
15:11	Reserved.
10:0	<b>s0tos5entmr</b> . Read-write. Reset: 055h. T12, timer for DebouncedPwrGood=1 to S5TilePwrMux=1, PwrGoodOsc=0 to xSLP_S3_/xSLP_S5_=0

**PMx0000002A (FCH::PM::s5tos0entmr)**

Read-write.

\_aliasHOST; PMx0000002A; PM=FED8\_0300h

Bits	Description
15:11	Reserved.
10:0	<b>s5tos0entmr</b> . Read-write. Reset: 010h. T3, timer for DebouncedPwrGood=1 to S5_ToAllEn=1

**PMx0000002C (FCH::PM::slprsttmr)**

Read-write.

\_aliasHOST; PMx0000002C; PM=FED8\_0300h

Bits	Description
15:11	Reserved.
10:0	<b>slprsttmr</b> . Read-write. Reset: 080h. T13, Timer for S3/S5 Sleep Reset pulse width

**PMx0000002E (FCH::PM::overridedevrsttmr)**

Read-write.

\_aliasHOST; PMx0000002E; PM=FED8\_0300h

Bits	Description
15:12	Reserved.
11:8	<b>overriderstbtmr</b> . Read-write. Reset: 0h. Override Default RstBTmr for device resets
7:4	<b>overriderefclkoktmr</b> . Read-write. Reset: 0h. Override Default RefClkOkTmr for device resets
3:0	<b>overridepwrrstbtmr</b> . Read-write. Reset: 0h. Override Default PwrRstBTmr for device resets

**PMx00000030 (FCH::PM::overridepwrrstbasrttmr)**

Read-write.

\_aliasHOST; PMx00000030; PM=FED8\_0300h

Bits	Description
15:11	Reserved.
10:0	<b>overridepwrrstbasrttmr</b> . Read-write. Reset: 000h. Override Default PwrRstBASrtTmr for device resets

**PMx00000032 (FCH::PM::overriderefclkokdsrttmr)**

Read-write.

\_aliasHOST; PMx00000032; PM=FED8\_0300h

Bits	Description
15:11	Reserved.
10:0	<b>overriderefclkokdsrttmr</b> . Read-write. Reset: 000h. Override Default RefClkOkDsrtTmr for device resets

**PMx00000034 (FCH::PM::overrideenable)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; PMx00000034; PM=FED8\_0300h

Bits	Description
31:0	<b>overrideenable</b> . Read-write. Reset: 0000_0000h. Enable DevRst block timing override. Each bit enables the override for a device's DevRst block. When the bit is 1, the corresponding device's DevRst block will use the timing defined in PM_Reg:2E~33h, instead of the original hardwired values.

**PMx00000038 (FCH::PM::devrst\_rstbonly)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; PMx00000038; PM=FED8\_0300h

Bits	Description
31:0	<b>devrst_rstbonly</b> . Read-write. Reset: 0000_0000h. Each bit is corresponding to a device's DevRst block. This is for debug purpose. When a bit is set 1, the corresponding device's PwrRsB and RefClkOk will be identical with RstB. When the bit is 0, the device's resets have default behaviors. Please refer to FCH_AOAC_Design_Specification for detail reset sequence and timing.

**PMx0000003C (FCH::PM::s0resettmr)**

Read-write.

\_aliasHOST; PMx0000003C; PM=FED8\_0300h

Bits	Description
15:12	Reserved.
11:0	<b>s0resettmr</b> . Read-write. Reset: 800h. Timer value for S0ResetB

**PMx0000003E (FCH::PM::pwrrstcnfg)**

Read-write. Reset: E601h.

\_aliasHOST; PMx0000003E; PM=FED8\_0300h

Bits	Description
15:8	<b>osc16usinterval</b> . Read-write. Reset: E6h. In decimal 230 (or 16us) by default. It is used to set the Osc16Us counter limit. It is the base timer tick to increment various reset counter. Whenever we should enter into S0i3, SMU FW can change Osc16UsInterval to a smaller value. Upon S0i3 exit, it can change it back to the default value.
7	Reserved.
6	<b>keep_warmrst_bf_mp1_ack</b> . Read-write. Reset: 0. <b>Description:</b> 0: warmrst sequence not wait MP1_Ack 1: warmrst sequence wait MP1_Ack indefinitely
5:3	Reserved.
2	<b>secondary_settings_enable</b> . Read-write. Reset: 0. Secondary settings enable
1	<b>en_mp1_warmresetack2</b> . Read-write. Reset: 0. <b>Description:</b> Chiken bit to select the gated iOsc16Us. 1: select the gated iOsc16Us by warm_rst_block_noncpu_rst which synced to osc clock domain. 0: select the original iOsc16Us.
0	<b>encpuwaitdev</b> . Read-write. Reset: 1. <b>Description:</b> 0: PwrRst block doesn't wait for non-D3 devices RstB de-assertion before it de-asserts CpuRstB. 1: PwrRst block will wait until all non-D3 devices RstB are de-asserted, and then de-assert CpuRstB. Note: "non-D3 devices" refer to those devices that are not currently in D3.

**PMx00000040 (FCH::PM::espiintrctrl)**

Read-write. Reset: 00FF\_FFFFh.

\_aliasHOST; PMx00000040; PM=FED8\_0300h

Bits	Description
31:24	Reserved.
23:0	<b>espidvintmask</b> . Read-write. Reset: FF_FFFFh. <b>Description:</b> SW can set these bits to mask of eSPI Device IRQ23~0. 1: Mask off the interrupt 0: No mask



**PMx00000044 (FCH::PM::boottimeren)**

Reset: 1800\_0000h.

\_aliasHOST; PMx00000044; PM=FED8\_0300h

Bits	Description
31	<b>boottmrdisable.</b> Read-write. Reset: 0. Set to 1 to stop boot timer. Once set it to 1, timer will reset back to 0. When this bit is set back to 0, timer starts counting from 0. This bit will clear itself on any reset (this bit is non-sticky).
30	<b>failbootrststs.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: Boot timer has not been fired. 1: Boot timer has been fired. Write 1 to clearit to 0.
29	<b>expireboottmr.</b> Read-write. Reset: 0. Set to 1 to force boot timer to expire then, NB PwrGood can be asserted.
28	<b>boottmrstopongalink.</b> Read-write. Reset: 1. Set to 1 to stop boot timer when FCH observes the good boot after PCI reset.
27	<b>boottmrfuncen.</b> Read-write. Reset: 1. <b>Description:</b> 0: Disable boot timer function. 1: Enable boot timer function. The boot timer will start to count down and toggle NBPwrGood after 1.26s ( $2^{24}/(66.6M/5)$ ) if SW has not set the bit 29 to 1 after PCI reset or resuming from S3/S4/S5. This bit is persistent through any reset or sleep (this bit is sticky).
26:25	<b>boottmrdivsel.</b> Read-write. Reset: 0h. <b>Description:</b> Select the Fail Boot Timer limit. 00b: 1.26 Second, using Divide_by_5, default 01b: 2.26 Second, using Divide_by_9 10b: 3.26 Second, using Divide_by_13 11b: 4.26 Second, using Divide_by_17
24:0	<b>failboottimer.</b> Read-write. Reset: 000_0000h. <b>Description:</b> Read Only. The counter of APU Boot timer (66.6MHz/5), which starts counting when all of the the following conditions are met: Bit31 of this register =0 Bit27 of this register =1 PCI reset is not asserted. Set bit31=1 will stop the timer and reset timer back to 0, it will count from 0 again if bit31 is set back to 0, bit31 is a non-sticky bit and will get clear to 0 on any reset. Set bit27=0 will disable timer permanently because bit27 is a sticky bit. Bit27 will reset back to 1 if there is a S5 power loss or it can be written back to 1 by software. The timer itself cannot be written directly. It will reset back to 0 whenever bit31=1 or bit27=0 or a reset has occurred.

**PMx00000048 (FCH::PM::pgpwrendly)**

Read-write. Reset: 5935\_2B21h.

\_aliasHOST; PMx00000048; PM=FED8\_0300h

Bits	Description
31:24	<b>pg1pwrdownldlytmr.</b> Read-write. Reset: 59h. PG1 Power Down delay timer
23:16	<b>xhcpwrendlytmr.</b> Read-write. Reset: 35h. XhcPwrEn delay timer
15:8	<b>pg2pwrendlytmr.</b> Read-write. Reset: 2Bh. PG2PwrEn delay timer
7:0	<b>pg1apwrendlytmr.</b> Read-write. Reset: 21h. PG1aPwrEn delay timer

**PMx0000004C (FCH::PM::i2cinputthreshold)**

Read-write. Reset: 493F\_0000h.

\_aliasHOST; PMx0000004C; PM=FED8\_0300h

Bits	Description
31:24	<b>gbepwrendlytmr.</b> Read-write. Reset: 49h. GbePwrEn delay timer
23:16	<b>otgpwrendlytmr.</b> Read-write. Reset: 3Fh. OtgPwrEn delay timer
15:6	Reserved.
5	<b>i2c5inputthresholdhi.</b> Read-write. Reset: 0. <b>Description:</b> Configure I2C5 input threshold 0: Low threshold 1: High threshold
4	<b>i2c4inputthresholdhi.</b> Read-write. Reset: 0. <b>Description:</b> Configure I2C4 input threshold 0: Low threshold 1: High threshold
3	<b>i2c3inputthresholdhi.</b> Read-write. Reset: 0. <b>Description:</b> Configure I2C3 input threshold 0: Low threshold 1: High threshold
2	<b>i2c2inputthresholdhi.</b> Read-write. Reset: 0. <b>Description:</b> Configure I2C2 input threshold 0: Low threshold 1: High threshold
1	<b>i2c1inputthresholdhi.</b> Read-write. Reset: 0. <b>Description:</b> Configure I2C1 input threshold 0: Low threshold 1: High threshold
0	<b>i2c0inputthresholdhi.</b> Read-write. Reset: 0. <b>Description:</b> Configure I2C0 input threshold 0: Low threshold 1: High threshold

**PMx00000050 (FCH::PM::apupllctrl)**

Read-write. Reset: 3F3E\_0000h.

\_aliasHOST; PMx00000050; PM=FED8\_0300h

Bits	Description
31:24	<b>apupllrstbtmr.</b> Read-write. Reset: 3Fh. Configure the PllLock assertion time. Unit is 16usec.
23:16	<b>apupllpwrrstbtmr.</b> Read-write. Reset: 3Eh. Configure the PllRstB deassertion time. Unit is 16usec.
15:2	Reserved.
1	<b>vidchg_use_newscheme.</b> Read-write. Reset: 0. <b>Description:</b> 0: old scheme, 8xPciclk wide VIDCHG_req generated when wakeup from S0i20 or S0i21 1: new scheme, while in S0i20 or S0i21, VIDCHG_req assert =1 when receiving wakeup signal, VIDCHG_req deassert =0 after receiving VIDCHG_ack =1, VIDCHG_ack will deassert =0 after seeing VIDCHG_req =0
0	<b>apupllctrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable 1: Eable APU to power down CGPLL(S0I21) or slow down clock (S0I20) When this bit is 1, ApuPllOffReq or S0I20_req bit in AOAC_Reg0x94 can be written by APU (or to be more precise, SMU) to power down the CGPLL or slow down clock.

**PMx00000054 (FCH::PM::serialirqconfig)**

Read-write. Reset: 0000h.

\_aliasHOST; PMx00000054; PM=FED8\_0300h

Bits	Description
15	Reserved.
14	<b>pmx54bit7_nonsticky.</b> Read-write. Reset: 0. <b>Description:</b> 1: bit[7] is non-Sticky 0: bit[7] is Sticky
13	<b>softwareirq12.</b> Read-write. Reset: 0. <b>Description:</b> 1: IRQ12 input is high 0: IRQ12 input is low
12	<b>softwareirq1.</b> Read-write. Reset: 0. <b>Description:</b> 1: IRQ1 input is high 0: IRQ1 input is low
11	Reserved.
10	<b>undo_ser_irq_change.</b> Read-write. Reset: 0. Not used.
9	<b>legacy_dis.</b> Read-write. Reset: 0. Not used.
8	<b>serialirq_cntrl.</b> Read-write. Reset: 0. Not used.
7	<b>serialirqenable.</b> Read-write. Reset: 0. <b>Description:</b> Setting this bit to 1 enable the serial IRQ function. When bit[14]=0, this bit is stick, default. When bit[14]=1, this bit is non-sticky
6	<b>serirqmode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Continuous mode 1: Active (quiet) mode
5:2	<b>numserirqbits.</b> Read-write. Reset: 0h. <b>Description:</b> Total number of serial IRQ's = 17 + NumSerIrqbits 0: 17 serial IRQ's (15 IRQ, SMI#, + IOCHK#) 1: 18 serial IRQ's (15 IRQ, SMI#, IOCHK#, INTA#) ... 15: 32 serial IRQ's The serial IRQ can support 15 IRQ#, SMI#, IOCHK#, INTA#, INTB#, INTC#, and INTD#. When serial SMI# is used, BIOS will need to check SIO (or device that generates serial SMI#) for status.
1:0	<b>numstartbits.</b> Read-write. Reset: 0h. <b>Description:</b> This field defines the number of clocks in the start frame. Start Frame Width = 4 + 2 * NumStartBits

**PMx00000056 (FCH::PM::rtccontrol)**

Read-write. Reset: 1400h.

\_aliasHOST; PMx00000056; PM=FED8\_0300h

Bits	Description
15	<b>rtcdecodedis.</b> Read-write. Reset: 0. <b>Description:</b> When it is 1, acpismbus stop to decode port 0x70/71/72/73. Note that, the reg nmi_en is in 0x70 bit7, this bit also block by RtcDecodeDis. Firmware have to config nmi_en before disable RTC decoding.
14	<b>extrartccmosen.</b> Read-write. Reset: 0. When it is 1, SW can access the extra 16 bytes of RTC CMOS RAM.
13	<b>altcmosmapen.</b> Read-write. Reset: 0. When enabled, bank 1 of CMOS RAM is changed. Index 00:0D will still return the time and alarm settings. Index 0E:7F will return the absolute offset 8E:FF.
12	<b>centuryen.</b> Read-write. Reset: 1. Enable RTC Century support.
11	<b>mask_rtc_clk_out.</b> Read-write. Reset: 0. Set to1 to disable RtcClk output.
10	<b>rtcclkdrive.</b> Read-write. Reset: 1. <b>Description:</b> 0: HIGHDRIVE tied low for RtcClkOut pad 1: HIGHDRIVE tied high for RtcClkOut pad
9:7	Reserved.
6	<b>rtc_test_en.</b> Read-write. Reset: 0. <b>Description:</b> This bit is for simulation only. Please do not change its value in real silicon. 0: Normal speed for prescaler 1: We replace 8KHz by OSC clock in prescaler. As the result, the prescaler 8KHz clock and all the clock divided from it can be speed up.
5	Reserved.
4	<b>rtcprotectc0_cf.</b> Read-write. Reset: 0. When set, RTC RAM index C0:CFh will be locked from read/write. This bit can only be written once.
3	<b>rtcprotectd0_df.</b> Read-write. Reset: 0. When set, RTC RAM index D0:DFh will be locked from read/write. This bit can only be written once.
2	<b>rtcprotecte0_ef.</b> Read-write. Reset: 0. When set, RTC RAM index E0:EFh will be locked from read/write. This bit can only be written once.
1	<b>rtcprotectf0_ff.</b> Read-write. Reset: 0. When set, RTC RAM index F0:FFh will be locked from read/write. This bit can only be written once.
0	<b>rt_cprotect38_3f.</b> Read-write. Reset: 0. When set, RTC RAM index 38:3Fh will be locked from read/write. This bit can only be written once.

**PMx00000058 (FCH::PM::vrt\_t1)**

Read-write. Reset: 01h.

\_aliasHOST; PMx00000058; PM=FED8\_0300h

Bits	Description
7:0	<b>vrt_t1.</b> Read-write. Reset: 01h. To conserve power, the RTC battery is sampled periodically for checking its state of health. VRT_T1 and VRT_T2 make up the interval of the checking. When VRT_Enable is high, the battery is being sampled. When VRT_enable is low, the battery is not being sampled. This register defines the time of VRT enable being high for RTC battery monitor circuit, in milliseconds.

**PMx00000059 (FCH::PM::vrt\_t2)**

Read-write. Reset: FFh.

\_aliasHOST; PMx00000059; PM=FED8\_0300h

Bits	Description
7:0	<b>vrt_t2.</b> Read-write. Reset: FFh. This register defines the time of VRT enable being low for the RTC battery monitor circuit, in 4 ms increments.

**PMx0000005A (FCH::PM::intrudercontrol)**

Read-write. Reset: 11h.

\_aliasHOST; PMx0000005A; PM=FED8\_0300h

Bits	Description
7	Reserved.
6	<b>cmoserasests.</b> Read-write. Reset: 0. Indicate that a CMOS Erase has been occurred.
5	<b>cmoseraseclr.</b> Read-write. Reset: 0. Write to 1 to clear CMOS Erase status.
4	<b>cmoserasedis.</b> Read-write. Reset: 1. Set to 1 to disable CMOS Erase.
3	Reserved.
2	<b>intruderalertsts.</b> Read-write. Reset: 0. <b>Description:</b> Read Only. The status bit will be set to 1 if an Intruder alter event (pad-IntruderAlert#=0) has occurred . Software need to set bit[1] to clear this status bit. To Arm®: set bit[0]=1, bit[1]=0 After Trigger: read bit[2] for status To re-Arm: set bit[0]=1, bit[1]=1 wait a short time set bit[1]=0
1	<b>intruderalertclr.</b> Read-write. Reset: 0. <b>Description:</b> Write 1 to clear the IntruderAlert status bit (bit[2]: IntruderAlertSts). Software need to write this bit 0 to enable pad-IntruderAlert#=0 to set bit[2]: IntruderAlertSts
0	<b>intruderalertpuen.</b> Read-write. Reset: 1. <b>Description:</b> 1: enable internal pullup (>200K). Please noted that pullup is enabled when both bit[0]=1 and bit[2]=0, if pad-IntruderAlert# is '0' while bit[1]=0, bit[2] will be set, and internal pullup will be disabled to save power. 0: disable internal pullup, when bit[0]=0, internal pullup is disabled.

**PMx0000005B (FCH::PM::rtcshadow)**

Read-write.

RTC\_AIE is defined at RTC\_Reg:0Bh[bit5].

Note1: These four bits[7:4] don't have any default value. After power on, their values are undertermined. SW has to program PwrFailShadow to give them values.

\_aliasHOST; PMx0000005B; PM=FED8\_0300h

Bits	Description
7	<b>forcepwron.</b> Read-write. Reset: X. <b>Description:</b> 0: If RTC AIE = 1, will wakeup when RTC alarm fires after a power failure/resume. (See Note) 1: If RTC AIE =1, will force power on after power resumes regardless of Bit[5:4] setting. (See Note)
6	<b>powerstate.</b> Read-write. Reset: X. <b>Description:</b> Power state indicator. 0: Off 1: On
5:4	<b>pwrfailoption.</b> Read-write. Reset: XXb. <b>Description:</b> These two bits will determine how system should resume after a power failure. 00: Always offalways power off after power resumes 01: Always onalways power on after power resumes 10: Always offalways power off after power resumes 11: Use previousresume to same setting when power fails
3:0	<b>pwrfailshadow.</b> Read-write. Reset: 0h. Writing to these four bits will set the value onto bits [7:4]. Software should always set bit 2 = 1.

**PMx0000005C (FCH::PM::llbcntrl)**

Read-write. Reset: 00h.

\_aliasHOST; PMx0000005C; PM=FED8\_0300h

Bits	Description
7:3	Reserved.
2	<b>allowwakes3en.</b> Read-write. Reset: 0. Set to 1 to allow LLB# as wake event in S3.
1	<b>useaswakeen.</b> Read-write. Reset: 0. Set to 1 to treat LLB# as wake event.
0	<b>blockwakeen.</b> Read-write. Reset: 0. Set 1 to block wake event if LLB# is asserted. But if UseAsWakeEn and AllowWakeS3En are all set to 1, LLB# and other wake events can wake the system up from S3.

**PMx0000005D (FCH::PM::rtcgpio)**

Read-write.

\_aliasHOST; PMx0000005D; PM=FED8\_0300h

Bits	Description
7	<b>ortclkpub.</b> Read-write. Reset: 0. <b>Description:</b> Pull up control for RTCCLK pad 0: Enable pull up 1: Disable pull up
6:2	Reserved.
1	<b>rtcirqgpioenb.</b> Read-write. Reset: 1. RtcIrq Gpio Output Enable
0	<b>rtcirqgpioout.</b> Read-write. Reset: 0. RtcIrq Gpio Output

**PMx0000005E (FCH::PM::rtcextindex)**

Read-write.

\_aliasHOST; PMx0000005E; PM=FED8\_0300h

Bits	Description
7:0	<b>index.</b> Read-write. Reset: XXXXXXXXb. Specify the offset of RTC Extended Registers to be read/written from PM_REG:5Fh

**PMx0000005F (FCH::PM::rtcextdata)**

Read-write.

\_aliasHOST; PMx0000005F; PM=FED8\_0300h

Bits	Description
7:0	<b>data.</b> Read-write. Reset: XXXXXXXXb. Read data or write data of RTC Extended Registers

**PMx00000060 (FCH::PM::acpipm1evtblk)**

Read-write. Reset: 0000h.

\_aliasHOST; PMx00000060; PM=FED8\_0300h

Bits	Description
15:2	<b>acpipm1evtblk.</b> Read-write. Reset: 0000h. These bits define the least significant byte of the 16 bit I/O range base address of the ACPI power management Event Block. Bit 2 corresponds to Addr[2] and bit 7 corresponds to Addr[7].
1:0	Reserved.

**PMx00000062 (FCH::PM::acpipm1cntblk)**

Read-write. Reset: 0400h.

\_aliasHOST; PMx00000062; PM=FED8\_0300h

Bits	Description
15:1	<b>acpipm1cntblk.</b> Read-write. Reset: 0200h. These bits define the least significant byte of the 16 bit I/O base address of the ACPI power management Control block. Bit 1 corresponds to Addr[1] and bit 7 corresponds to Addr[7].
0	Reserved.

**PMx00000064 (FCH::PM::acpipmtmrblk)**

Read-write. Reset: 0000h.

\_aliasHOST; PMx00000064; PM=FED8\_0300h

Bits	Description
15:1	<b>acpipmtmrblk.</b> Read-write. Reset: 0000h. These bits define the least significant byte of the 16 bit I/O base address of the ACPI power management Timer block. Bit 1 corresponds to Addr[1] and bit 7 corresponds to Addr[7].
0	Reserved.

**PMx00000066 (FCH::PM::p\_cntblk)**

Read-write. Reset: 0000h.

\_aliasHOST; PMx00000066; PM=FED8\_0300h

Bits	Description
15:0	Reserved.

**PMx00000068 (FCH::PM::acpigpe0blk)**

Read-write. Reset: 0000h.

\_aliasHOST; PMx00000068; PM=FED8\_0300h

Bits	Description
15:2	<b>acpigpe0blk.</b> Read-write. Reset: 0000h. These bits define the least significant byte of the 16 bit I/O base address of the ACPI power management General Purpose Event block. Bit 2 corresponds to Addr[2] and bit 7 corresponds to Addr[7]. Addr[1:0] are ignored because this register block is 4 byte long.
1:0	Reserved.

**PMx0000006A (FCH::PM::acpismicmd)**

Read-write. Reset: 00B0h.

\_aliasHOST; PMx0000006A; PM=FED8\_0300h

Bits	Description
15:0	<b>acpismicmd.</b> Read-write. Reset: 00B0h. These bits define the least significant byte of the 16 bit I/O base address of the ACPI SMI Command block. Bit 0 corresponds to Addr[0] and bit 7 corresponds to Addr[7]. The address is required to be WORD-aligned (Addr[0]=0)

**PMx0000006E (FCH::PM::acpipmacntblk)**

Read-write. Reset: 0000h.

\_aliasHOST; PMx0000006E; PM=FED8\_0300h

Bits	Description
15:0	<b>acpipmacntblk.</b> Read-write. Reset: 0000h. These bits define the most significant byte of the 16 bit I/O base address. Bit 0 corresponds to Addr[8] and bit 7 corresponds to Addr[15].

**PMx00000070 (FCH::PM::mp1\_warmresetack\_wlan\_clk\_req)**

Read-write. Reset: 3F3E\_0170h.

\_aliasHOST; PMx00000070; PM=FED8\_0300h

Bits	Description
31:24	<b>wlanpllrbtmtmr.</b> Read-write. Reset: 3Fh. Configure the PllLock assertion (0->1) time after PllRstB deasserts (0->1). Unit is 16usec.
23:16	<b>wlanpllpwrrstbtmtmr.</b> Read-write. Reset: 3Eh. Configure the PllRstB deassertion (0->1) time. Unit is 16usec.
15	<b>wlanpllctrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable 1: Enable WLAN to Control CGPLL When this bit is 1 and system in S0I21/S0I3/S3/S4/S5, wlan_clk_req could control FCH_APU_PllRstB/FCH_APU_PllLock which go to CGPLL
14	<b>newwarmrstblockdisable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Device reset will not be blocked by MP1_WarmResetAck handshake. 1: Device reset will be blocked by MP1_WarmResetAck handshake.
13:3	<b>wdt_warmresetack_tmr.</b> Read-write. Reset: 02Eh. <b>Description:</b> Time out value of MP1_Watchdog timer, this time is using 16.2us Osc clock to do increment. default=0x2E After append 2'b11, the Timer is 187. (16.2us*187=3.029ms), max=16.2us*8191=132.694ms
2	<b>en_slv_cpurst.</b> Read-write. Reset: 0. when set to 1, Master FCH will do reset when Slave FCH asserts CpuRstB=0. Slave FCH will do Warm Reset for sync_flood.
1	<b>en_wdt_warmresetack.</b> Read-write. Reset: 0. when bit[0]=1, set to 1 to enable MP1_Watchdog Timer for WarmResetAck handshake, if MP1_WarmResetAck rising edge not detected before MP1_Watchdog Timer time out, Cold reset will be generated.
0	<b>en_mp1_warmresetack.</b> Read-write. Reset: 0. set to 1 to enable MP1_WarmResetAck handshake, all FCH controlled Devices Reset, except PLL and APU, will be blocked till MP1_WarmResetAck rising edge or MP1_Watchdog Timer time out.



**PMx00000074 (FCH::PM::acpiconfig)**

Read-write. Reset: 1000\_0140h.

\_aliasHOST; PMx00000074; PM=FED8\_0300h

Bits	Description
31	<b>delay_gpp_off_time.</b> Read-write. Reset: 0. <b>Description:</b> When BP_PWR_GOOD is used as RESET button, once pressed, PCI GPP clock will be stopped right away while PciRST# will be active (=0) some time later, this bit is used to meet customer's request to have PCI GPP clock running after PciRST# active. This bit is reset by RsmRstB only. 1: PCI BPP clock keep running for some time after PciRST#=0 0: PCI BPP clock stopped right away when BP_PWR_GOOD=0
30	<b>use_gatedalinkclk.</b> Read-write. Reset: 0. <b>Description:</b> 0: Some logics in LPC are using free-run AlinkClk, same as before. 1: Some logics in LPC are using gated AlinkClk to save power.
29	<b>rtcwakealarm.</b> Read-write. Reset: 0. Set to 1 to only rtc alarm to wake up the system.
28	<b>pcieeventmap.</b> Read-write. Reset: 1. Set to 1 to route pme message from APU to gevent 24, Hotplug message from APU to gevent 7..
27	<b>wakepinasgevent.</b> Read-write. Reset: 0. Set to 1 to treat Wake# pin as Gevent input.
26	<b>pcie_wak_intr_dis.</b> Read-write. Reset: 0. Set to 1 to disable interrupt from Pcie_wak_sts
25	<b>pcie_wak_mask.</b> Read-write. Reset: 0. Set to 1 to disable PCIE_WAK_STS and PCIE_WAK_DIS function. This is used for supporting ACPI 3.0 specification. If ACPI 3.0 is not supported, this bit should be left as 1
24	<b>pcienative.</b> Read-write. Reset: 0. Setting to 1 will block PCIe® GPP PME message and HotPlug message from generating SCI. This is used for supporting ACPI 3.0 specification. If ACPI 3.0 is not supported, this bit should be left as 0
23	<b>rst_usb_s5.</b> Read-write. Reset: 0. Write '1' to make Cpl_VDDCR_S5_RESETEn=0 to reset USB,MP2, etc. Need to write '0' to deassert Cpl_VDDCR_S5_RESETEn=1. This bit is not for normal operation, it is to be used for recovery when something went wrong.
22	<b>usersmureset.</b> Read-write. Reset: 0. <b>Description:</b> 0: fch_rsmu_hard_resestb is not used to reset fch_rsmu, SDP, AB. 1: fch_rsmu_hard_resestb is used to reset fch_rsmu, SDP, AB.
21	<b>mask_usb_s5_rst.</b> Read-write. Reset: 0. Set to '1' to mask oDevAllRstB to gen Cpl_VDDCR_S5_RESETEn=0 to reset USB/MP2/..., default=0
20	<b>en_df_intrwake.</b> Read-write. Reset: 0. <b>Description:</b> 0: disable FCH_DF_IntrWake in s5_misc.vpp 1: enable FCH_DF_IntrWake in s5_misc.vpp
19	Reserved.
18	<b>en_sync_flood.</b> Read-write. Reset: 0. <b>Description:</b> 0: disable SYNC_FLOOD 1: enable SYNC_FLOOD
17	<b>en_shutdown_msg.</b> Read-write. Reset: 0. <b>Description:</b> 0: disable SHUTDOWN message 1: enable SHUTDOWN message
16	<b>sw_s5pwrmux.</b> Read-write. Reset: 0. <b>Description:</b> When PM_regx74[15]=0, this bit control oS5PwrMux to USB. This bit will be reset to 0 by PllLock=0, thus it will be reset to 0 on any S5/S3 entry. 0: USB is powered by S5-domain VDDCR_SOC_S5 (bypass to S5 rail mode) 1: USB is powered by CLDO Vout (regulation mode, to be used in S0)
15	<b>sw_s5pwrmux_override_n.</b> Read-write. Reset: 0.

	<b>Description:</b> 0: use PM_regx74[16] to control oS5PwrMux to USB. 1: use iPlllLock to control oS5PwrMux to USB. When iPlllLock is unlock (=0), USB is powered by S5-domain VDDCR_SOC_S5 (bypass to S5 rail mode) When iPlllLock is locked (=1), USB is powered by CLDO Vout (regulation mode, to be used in S0)
14	<b>sel_smu_thermtrip.</b> Read-write. Reset: 0. <b>Description:</b> 0: use synced SMU_THERMTRIP to drive pad BP_THERMTRIP_L 1: use SMU_THERMTRIP directly to drive pad BP_THERMTRIP_L
13	<b>sel_pwrgd_pad.</b> Read-write. Reset: 0. <b>Description:</b> 0: SPI_FBCLK_mux will use debounced PwrGood PAD input (~10ms) as ResetB to avoid glitch. 1: SPI_FBCLK_mux will use PwrGood PAD input as ResetB
12	<b>lpc_rst_dis.</b> Read-write. Reset: 0. <b>Description:</b> assign IO_LPC_RST_L_IN_iA = LPC_RESETB & !LPC_RST_Dis It is used to gate LPC_RST_L. default eanbled.
11	<b>s5_lpcpinmode.</b> Read-write. Reset: 0. Co-work with above bit in the above logic.
10	<b>s5_lpcpinmodesel.</b> Read-write. Reset: 0. <b>Description:</b> new_EMMC_V18_EN = !Type2StrapDone    (PMC_RomStrap_S5 ? 1'b0 : S5_LpcPinModeSel ? S5_LpcPinMode : EMMC_V18_EN) When the result is high, the pad power is 1.8v When the result is low, the pad power is 3.3v. LPC need to work on 3.3v and EMMC need to work on 1.8v.
9	<b>acpireducedhwen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Acpi Fixed register interface is enabled 1: the decoding of Acpi Fixed registers and Sci are disabled. In addition wake function from pm1a is disabled as well.
8	<b>pwnbbtnen.</b> Read-write. Reset: 1. Set to 1 to enable power button support in pm1a block
7	<b>bios_rls.</b> Read-write. Reset: 0. Set to1 to generate SCI. Read always return 0
6	<b>maskarbdis.</b> Read-write. Reset: 1. Set to1 to disable ArbDis function in the ACPI register. ArbDis is not really used, but it still needs to be accessible by OS.
5	Reserved.
4	<b>tmr_en_en.</b> Read-write. Reset: 0. Set to 1 enable TMR_EN function in the standard ACPI register. This is designed as a backdoor for BIOS to control it.
3	<b>slpbtn_en_en.</b> Read-write. Reset: 0. Set to1 to enable SLPBTN_EN function in the standard ACPI register. This is designed as a backdoor for BIOS to control it
2	<b>rtc_en_en.</b> Read-write. Reset: 0. Set to1 to enable RTC_EN function in the standard ACPI register. This is designed as a backdoor for BIOS to control it.
1	<b>gbl_en_en.</b> Read-write. Reset: 0. Set to 1 to enable GBL function in the standard ACPI PmControl register.
0	<b>decen_acpi.</b> Read-write. Reset: 0. Set to 1 to enable decoding of the standard ACPI registers

**PMx00000078 (FCH::PM::wakeioaddr)**

Read-write. Reset: 0000\_FFFFh.

\_aliasHOST; PMx00000078; PM=FED8\_0300h

Bits	Description
31:17	Reserved.
16	<b>mp1_wdt_cnt_clr.</b> Read-write. Reset: 0. <b>Description:</b> Write one clear. Write this bit to one, it will clear the MP1_WDT_cnt in PwrRst.
15:0	<b>wakeibaseaddress.</b> Read-write. Reset: FFFFh. The register specifies the wake I/O address. Any I/O write to the I/O address can cause APU to wake from C state. This is an obsolete function that is not used anymore

**PMx0000007E (FCH::PM::cstateen)**

Read-write. Reset: 0060h.

\_aliasHOST; PMx0000007E; PM=FED8\_0300h

Bits	Description
15:6	Reserved.
5	<b>k8c1etoc3en.</b> Read-write. Reset: 1. Set to 1 to put APU into C3 state in C1e state.
4	<b>k8c1etoc2en.</b> Read-write. Reset: 0. Set to 1 to put APU into C2 state in C1e state.
3:0	Reserved.

**PMx00000080 (FCH::PM::breakevent)**

Read-write. Reset: 0026\_0000h.

\_aliasHOST; PMx00000080; PM=FED8\_0300h

Bits	Description
31:24	Reserved.
23:16	<b>scratchbit16_23.</b> Read-write. Reset: 26h.
15:5	Reserved.
4	<b>scratchbit4.</b> Read-write. Reset: 0. Scratch bit used by Software
3	<b>scratchbit3.</b> Read-write. Reset: 0. Scratch bit used by Software
2	<b>scratchbit2.</b> Read-write. Reset: 0. Scratch bit used by Software
1	Reserved.
0	<b>scratchbit0.</b> Read-write. Reset: 0. Scratch bit used by Software

**PMx00000088 (FCH::PM::cstatecontrol)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; PMx00000088; PM=FED8\_0300h

Bits	Description
31:11	Reserved.
10	<b>sw_sync_flood.</b> Read-write. Reset: 0. When PMx8c[27]=1, write this bit =1 will generate SW_sync_flood. Since this register is non-sticky, this bit will be cleared by PciRstB, thus software no need to write 0 after write 1 should this SW_sync_flood is enabled to generate Reset (PMx74[18]=1).
9:6	Reserved.
5	<b>slp_en.</b> Read-write. Reset: 0. Enable LDTSTOP# as an output
4:3	Reserved.
2	<b>dlyslpen.</b> Read-write. Reset: 0. Set to 1 to delay recognition of STPGNT# until there is no pending read in AB
1:0	Reserved.

**PMx0000008C (FCH::PM::bttncsicontrol)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; PMx0000008C; PM=FED8\_0300h

Bits	Description
31	<b>bttn_diswake_pressins0.</b> Read-write. Reset: 0. <b>Description:</b> 1: no wakeup when button pressed in S0, and software put system in S3 while button still pressed and then released in less than 4s 0: default, when button is pressed and released in less than 4s, if system is in S3/S5, system will always do wakeup.
30	<b>sci_bttm_fall_edge.</b> Read-write. Reset: 0. <b>Description:</b> 1: set PWRBTN_STS_reg to gen SCI at button falling edge (after 16ms debounce) 0: default, no SCI at button falling edge
29	<b>wlan_fp6_iso_en.</b> Read-write. Reset: 0. <b>Description:</b> 0: WLAN ISO off, S5 WLAN PMA to S5 WLAN digital signals are disconnected. 1: WLAN ISO on, S5 WLAN PMA to S5 WLAN digital signals are connected.
28	<b>smireg_use_s5rst.</b> Read-write. Reset: 0. <b>Description:</b> This bit is reset by RsmRstB only. 0: some registers inside SMI (acpi_event) module are reset by PciRstB 1: some registers inside SMI (acpi_event) module are reset by S5RstB
27	<b>sw_sync_flood_enable.</b> Read-write. Reset: 0. <b>Description:</b> 0: PMx88[10] cannot generate SW_sync_flood. 1: PMx88[10] can generate SW_sync_flood.
26	<b>dis_sci_wakeup_s0i3.</b> Read-write. Reset: 0. <b>Description:</b> 0: sci_wakeup can wake S0i3 1: sci_wakeup cannot wake S0i3
25	<b>dis_pic_irq_8_s0i3.</b> Read-write. Reset: 0. <b>Description:</b> 0: pic_irq_8 can wakeup S0i3 1: pic_irq_8 cannot wakeup S0i3
24	<b>acpialarm_nchk_tmren.</b> Read-write. Reset: 0. <b>Description:</b> 0: ACPIAlarm will check TMR_EN =1 1: ACPIAlarm will not check TMR_EN
23	<b>dis_smi_wakeup_s0i3.</b> Read-write. Reset: 0. <b>Description:</b> 0: smi_wakeup can wake s0i3 1: smi_wakeup cannot wake s0i3
22	<b>dis_nmi_wakeup_s0i3.</b> Read-write. Reset: 0. <b>Description:</b> 0: nmi_wakeup can wake s0i3 1: nmi_wakeup cannot wake s0i3
21	<b>dis_sci_wakeup_z8.</b> Read-write. Reset: 0. <b>Description:</b> 0: sci_wakeup can wake Z8 1: sci_wakeup cannot wake Z8
20	<b>dis_nmi_wakeup_z8.</b> Read-write. Reset: 0. <b>Description:</b> 0: nmi_wakeup can wake Z8 1: nmi_wakeup cannot wake Z8
19	<b>dis_smi_wakeup_z8.</b> Read-write. Reset: 0. <b>Description:</b> 0: smi_wakeup can wake Z8 1: smi_wakeup cannot wake Z8
18	<b>gpio_inten0_use_wakeen.</b> Read-write. Reset: 0. <b>Description:</b> 0: GPIO internal int_en[0] could be derived from ( wake_ctrl[2:0])   wake_ctrl_z) 1: GPIO internal int_en[0] could be derived from ( wake_ctrl[2:0])   wake_ctrl_z) qualified with its own system-in-low-power-state signal

17:0	Reserved.
------	-----------

**PMx00000090 (FCH::PM::spipadpupd)**

Read-write. Reset: 88A4\_0955h.

\_aliasHOST; PMx000000090; PM=FED8\_0300h

Bits	Description
31	<b>boot_timer_cold_rst.</b> Read-write. Reset: 1. When bit[31]=0, Boot Timer Reset is doing warm reset (toggle Cpu_ResetB only). By setting bit[31]=1, Boot Timer Reset will do cold reset (toggle both Cpu_ResetB and Cpu_PwrGood).
30	<b>dis_spi_pupd_sw.</b> Read-write. Reset: 0. <b>Description:</b> Set to 1 to disable all SPI PAD pullup/pulldown defined in this register. When we are in S3/S5, external S0VDD will be power-off, though SPI PADs are in S5, depends on board design, S0VDD may be connected to SPI PAD. To enable SPI ROM sharing with external EC, all pullup/pulldown need be disabled so external EC can drive SPI signals to access SPI ROM. Software can set bit[30]=1 without change bit [11:0] value before getting into S3/S5, after exit S3/S5, software need to reset bit[30]=0.
29	<b>dis_spi_pupd_hw.</b> Read-write. Reset: 0. If set to 1, when system is in S0i3 or S3 or S5, all SPI PAD pullup/pulldown defined in this register will be disabled.
28	<b>spi2_cs1_l_pulldown.</b> Read-write. Reset: 0. control pad SPI2_CS1 pulldown, 1: Enable 0: disable
27	<b>spi2_cs1_l_pullup.</b> Read-write. Reset: 1. control pad SPI2_CS1 pullup, 1: Enable 0: disable
26:24	<b>st_sx.</b> Read-write. Reset: 0h. <b>Description:</b> Read Only, indicate the current S-state value, for debug purpose only, as in S0i21/S0i3/S3/S4/S5, Host cannot read register value, 0x0: S0 state 0x1: S0i20 state 0x2: S0i21 state 0x3: S3 state 0x4: S0i3 state 0x5: S4/S5 state 0x6, 0x7: reserved
23	<b>spi1_hold_l_pulldown.</b> Read-write. Reset: 1. control pad SPI1_HOLD_L pulldown, 1: Enable 0: disable
22	<b>spi1_hold_l_pullup.</b> Read-write. Reset: 0. control pad SPI1_HOLD_L pullup, 1: Enable 0: disable
21	<b>spi1_wp_l_pulldown.</b> Read-write. Reset: 1. control pad SPI1_WP_L pulldown, 1: Enable 0: disable
20	<b>spi1_wp_l_pullup.</b> Read-write. Reset: 0. control pad SPI1_WP_L pullup, 1: Enable 0: disable
19	<b>spi1_cs1_l_pulldown.</b> Read-write. Reset: 0. control pad SPI1_CS1 pulldown, 1: Enable 0: disable
18	<b>spi1_cs1_l_pullup.</b> Read-write. Reset: 1. control pad SPI1_CS1 pullup, 1: Enable 0: disable
17	<b>bt_gpio_1p8v_receive.</b> Read-write. Reset: 0. <b>Description:</b> 0: GPIO7/GPIO8 are 3.3V receiver 1: GPIO7/GPIO8 are 1.8V receiver, when they used as BT receiver
16:12	Reserved.
11	<b>spi_clk_pulldown.</b> Read-write. Reset: 1. control pad SPI_CLK pulldown, 1: Enable 0: disable
10	<b>spi_clk_pullup.</b> Read-write. Reset: 0. control pad SPI_CLK pullup, 1: Enable 0: disable
9	<b>spi_cs1_pulldown.</b> Read-write. Reset: 0. control pad SPI_CS1 pulldown, 1: Enable 0: disable
8	<b>spi_cs1_pullup.</b> Read-write. Reset: 1. control pad SPI_CS1 pullup, 1: Enable 0: disable
7	<b>spi_hold_l_pulldown.</b> Read-write. Reset: 0. control pad SPI_HOLD_L pulldown, 1: Enable 0: disable
6	<b>spi_hold_l_pullup.</b> Read-write. Reset: 1. control pad SPI_HOLD_L pullup, 1: Enable 0: disable
5	<b>spi_wp_l_pulldown.</b> Read-write. Reset: 0. control pad SPI_WP_L pulldown, 1: Enable 0: disable
4	<b>spi_wp_l_pullup.</b> Read-write. Reset: 1. control pad SPI_WP_L pullup, 1: Enable 0: disable
3	<b>spi_do_pulldown.</b> Read-write. Reset: 0. control pad SPI_DO pulldown, 1: Enable 0: disable
2	<b>spi_do_pullup.</b> Read-write. Reset: 1. control pad SPI_DO pullup, 1: Enable 0: disable
1	<b>spi_di_pulldown.</b> Read-write. Reset: 0. control pad SPI_DI pulldown, 1: Enable 0: disable
0	<b>spi_di_pullup.</b> Read-write. Reset: 1. control pad SPI_DI pullup, 1: Enable 0: disable

**PMx00000094 (FCH::PM::cstatetiming0)**

Read-write. Reset: 0010\_0000h.

\_aliasHOST; PMx00000094; PM=FED8\_0300h

**Bits Description**

31:0 Reserved.

**PMx0000009A (FCH::PM::firmwarescratch)**

Read-write. Reset: 0000h.

\_aliasHOST; PMx0000009A; PM=FED8\_0300h

**Bits Description**15:0 **scratch16**. Read-write. Reset: 0000h. This is a place holder to keep 16 bits for other IP or Software.**PMx000000A0 (FCH::PM::messagecstate)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; PMx000000A0; PM=FED8\_0300h

**Bits Description**

31 **checkcoreiddis**. Read-write. Reset: 0.  
**Description:** 1: Ignore Core Id check when MultiCoreEnbit set 1.  
 0: Enable Core Id check when MultiCoreEnbit set 1.

30:21 **pmxa0reserved**. Read-write. Reset: 000h.20:16 **tmrseloverride**. Read-write. Reset: 00h.

**Description:** This is to be used with bit TimerTickChgMsgEn. In case FCH auto-timer detection logic is not functioning properly, one can use these bits to override the logic and force the logic to monitor the specific timer.  
 Bit 0 When set, use HPET  
 Bit 1 When set, use RTC  
 Bit 2 if HPET is selected, setting bit 2 will force the logic to monitor HPET timer 0.  
 Bit 3 if HPET is selected, setting bit 3 will force the logic to monitor HPET timer 1.  
 Bit 4 if HPET is selected, setting bit 4 will force the logic to monitor HPET timer2.

15:11 Reserved.

10 **clkintrtagen**. Read-write. Reset: 0. When enabled, FCH will mark the periodic timer interrupt.

9:2 Reserved.

1 **timertickchgmsgen**. Read-write. Reset: 0. When enabled, FCH will send a message to CPU indicating the latest periodic timer interval. FCH will automatically determine which timer (PIT, RTC, or HPET) is being used.

0 **obattmodechgmsgen**. Read-write. Reset: 0. When enabled, FCH will automatically send a message to CPU indicating the power mode (AC vs battery). In addition, every time it is changed, FCH will generate a message to indicate the update.

**PMx000000A4 (FCH::PM::miscdebug)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; PMx000000A4; PM=FED8\_0300h

**Bits Description**

31:3 Reserved.

2 **blockgeventsmi**. Read-write. Reset: 0.

**Description:** 1: Block GEVENT SMI  
 0: Unblock GEVEN SMI

1 **blocksci**. Read-write. Reset: 0.

**Description:** 1: Block SCI  
 0: Unblock SCI

0 Reserved.

**PMx000000A8 (FCH::PM::virtualwire)**

Read-write. Reset: 0023\_0085h.

\_aliasHOST; PMx000000A8; PM=FED8\_0300h

Bits	Description
31:27	Reserved.
26:16	<b>otherpwrrsttmr</b> . Read-write. Reset: 023h. Parametised the reset timing wait time for other devices.
15:11	Reserved.
10:0	<b>pg1pwrrsttmr</b> . Read-write. Reset: 085h. Parametised the reset timing wait time for devices in PG1.



**PMx000000B0 (FCH::PM::defertimetick\_obffenable)**

Read-write.

\_aliasHOST; PMx000000B0; PM=FED8\_0300h

Bits	Description
31	<b>sw_obff_override.</b> Read-write. Reset: 0. <b>Description:</b> 0: CpuActive/MemActive from Data Fabric are used for OBFF. 1: bit[30:29] are used for OBFF.
30	<b>sw_obff_cpuactive.</b> Read-write. Reset: 0. When bit[31]=Sw_obff_override=1, this bit will override CpuActive from Data Fabric to control OBFF operation when bit[11]=OBFFEn=1
29	<b>sw_obff_memactive.</b> Read-write. Reset: 0. When bit[31]=Sw_obff_override=1, this bit will override MemActive from Data Fabric to control OBFF operation when bit[11]=OBFFEn=1
28	<b>long10usdelay.</b> Read-write. Reset: 0. <b>Description:</b> Spec requires ACTIVE/OBFF -> IDLE to wait at least 10us, for other transition ACTIVE/IDLE -> OBFF or OBFF/IDLE -> ACTIVE, the delay is controlled by this bit. 0: use 3x400ns (400ns controlled by bit[6:2]) 1: use 10us (10us controlled by bit[21:12])
27	<b>cpuactive.</b> Read-write. Reset: X. <b>Description:</b> Read Only. State of Internal signal CpuActive that from Data Fabric
26	<b>memactive.</b> Read-write. Reset: X. <b>Description:</b> Read Only. State of Internal signal MemActive that from Data Fabric.
25:24	<b>obff_state.</b> Read-write. Reset: 0h. <b>Description:</b> Read Only. Internal OBFF state machine states. 00: IDLE (when bit[11]=OBFFEn=0, OBFF_state is IDLE) 01: OBFF 11: ACTIVE
23	<b>obff_transition_done.</b> Read-write. Reset: 1. <b>Description:</b> Read Only. 1: OBFF transition is done 0: OBFF transition is going on.
22	<b>en_obff_blk_wake.</b> Read-write. Reset: 0. <b>Description:</b> 0: input of pad_WAKE_L will not be blocked from generating event when OBFF state machine output pattern. 1: input of pad_WAKE_L will be blocked from generating event when OBFF state machine output pattern of "IDLE/ACTIVE -> OBFF" or "IDLE/OBFF -> ACTIVE", and will be enabled when OBFF state machine output pattern of "OBFF/ACTIVE -> IDLE)
21:12	<b>cnt10usper48m.</b> Read-write. Reset: 1E0h. <b>Description:</b> Number of 48MHz clock for 10us delay, used for spec requirement "when platform enter CPU Active or OBFF, platform should not return to Idle in less than 10us". Software can change this value to adjust the delay. Default=0x1e0=480*(1sec/48M)=10us
11	<b>obffen.</b> Read-write. Reset: 0. Set to 1 to enable OBFF function which toggling WAKE#
10:8	<b>defertimetickvalue.</b> Read-write. Reset: 0h.

	<b>Description:</b> 000: No skipping 001: Skip 1 timer tick 010: Skip 2 timer ticks 011: Skip 3 timer ticks 100: Skip 4 timer ticks 101: Skip 5 timer ticks 110: Skip 6 timer ticks 111: Skip 7 timer ticks
7	<b>force_obff_blk_wake.</b> Read-write. Reset: 0. When set to 1, input of pad_WAKE_L will always be blocked from generating event.
6:2	<b>cnt400nsper48m.</b> Read-write. Reset: 14h. <b>Description:</b> Number of 48MHz (20.83ns) clock for 400ns delay, used for pulse width (both high and low pulse) of WAKE# protocol, software can change this value to adjust the WAKE# pulse width. Default=0x14=20*20.83 =417ns.
1	<b>forcetmrticken.</b> Read-write. Reset: 0. If bit 0 is set along with this bit and FCH has skipped a timer tick interrupt, FCH will immediately generate the timer tick interrupt upon C state exit
0	<b>defertimerticken.</b> Read-write. Reset: 0. When set, FCH will skip a number of timer tick interrupts based on the defined value in DeterTimeTickValue when CPU is in C state. When CPU is not in C state, FCH will not skip any timer tick interrupts.

**PMx000000B4 (FCH::PM::acpimiscdebug)**

Read-write. Reset: 0580\_0000h.

\_aliasHOST; PMx000000B4; PM=FED8\_0300h

Bits	Description
31:30	Reserved.
29:24	<b>tpreset1b.</b> Read-write. Reset: 05h. <b>Description:</b> Timing parameter used for S* -> S0 state transition. This determines the delay between CPU_STP# de-assertion and LPC_PD# de-assertion, in 8s increment with 8s uncertainty.
23	<b>intrtoggleonldtstp.</b> Read-write. Reset: 1. <b>Description:</b> Configure the behavior of "DMA/Interrupt indicator": When LdtStpB behaves as interrupt indicator, this bit configure the way it indicates interrupt: 0: The interrupt is signaled as a level signal. 1: The interrupt is signaled as a clock signal with 2us or 4us period depends on LDTSTPBTrSel.
22	<b>ldtstpbutoutputdis.</b> Read-write. Reset: 0. <b>Description:</b> Configure the behavior of "DMA/Interrupt indicator": Disable LdtStpB output: 0: LdtStpB output enabled 1: LdtStpB output disabled
21	<b>wakecstateinslp.</b> Read-write. Reset: 0. <b>Description:</b> Configure the behavior of "DMA/Interrupt indicator": 0: Do not wake C State before entering S State. 1: Wake C State before entering S State.
20	<b>mergeallowldtstpwithldtstp.</b> Read-write. Reset: 0. <b>Description:</b> Configure the behavior of "DMA/Interrupt indicator": Change the definition of pin DMAACTIVE_L (ALLOWLDTSTP). 0: Drive DMAACTIVE_L to low when there is pending upstream request. 1: Output 2us/4us toggling on DMAACTIVE_L (ALLOWLDTSTP) when there is pending interrupt request and not in C0 state to DMAACTIVE_L otherwise drive DMAACTIVE_L to low when there is pending upstream request
19	<b>intronldtstpben.</b> Read-write. Reset: 0. <b>Description:</b> Configure the behavior of "DMA/Interrupt indicator": Change the definition of LdtStpB pin to "interrupt indicator". 0: LdtStpB behaves as LdtStpB 1: LdtStpB behaves as interrupt indicator When this bit is 1 and there is an interrupt, LdtStpB pin will toggle in the rate defined in EcoBit1[0].
18	<b>ldtstpbtmrsl.</b> Read-write. Reset: 0. <b>Description:</b> Configure the behavior of "DMA/Interrupt indicator": Select 2us or 4us toggling on DMAACTIVE_L (ALLOWLDTSTP) or LDTSTPB_L according to Merge_Interrupt_Dma_Reg value. 0: 2us 1: 4us
17	Reserved.
16	<b>oallowldtstpasout.</b> Read-write. Reset: 0. <b>Description:</b> Control the input/output direction of pin AllowLdtStp. 0: (default) AllowLdtStp is input to FCH driven by APU. 1: AllowLdtStp as output pin to APU to indicate FCH traffic activities.
15:4	Reserved.
3	<b>vfidclrthroten.</b> Read-write. Reset: 0. Enable the fix for #SB02608 (Part 2)
2	<b>fidvidatom.</b> Read-write. Reset: 0. Enable the fix for #SB02608 (Part 4)
1	<b>fidrestart.</b> Read-write. Reset: 0. Enable the fix for #SB02608 (Part 3)

0	<b>fid_protect_en.</b> Read-write. Reset: 0. Set to 1 to skip C-state transition when FID/VID message is received concurrently.
---	---

**PMx000000B8 (FCH::PM::tpreset2)**

Read-write. Reset: 88h.

\_aliasHOST; PMx000000B8; PM=FED8\_0300h

Bits	Description
7:6	<b>clkgatecntrl.</b> Read-write. Reset: 2h. <b>Description:</b> These two bits control whether SMBUS module will allow clock gating to the internal 66Mhz core clock 00: Disable the clock gating function 01: Wait 16 clocks before allowing clock gating to the SMBUS module 10: Wait 64 clocks before allowing clock gating to the SMBUS module 11: Wait 256 clocks before allowing clock gating to the SMBUS module
5:0	<b>tpreset2.</b> Read-write. Reset: 08h. Timing parameter used for S* -> S0 state transitions. This register determines the LDTSTOP# deassertion delay in 8s increment with 8s uncertainty.

**PMx000000B9 (FCH::PM::lpcmiscdebug)**

Read-write. Reset: 00h.

\_aliasHOST; PMx000000B9; PM=FED8\_0300h

Bits	Description
7	Reserved.
6	<b>test_pwr_btn.</b> Read-write. Reset: 0. <b>Description:</b> This bit is used to speed up simulation 1'b1: debounce 16ms counter is 0x12 1'b0: debounce 16ms counter is 0x1ff
5:4	Reserved.
3	<b>clkrunisable.</b> Read-write. Reset: 0. Legacy DMA and serial IRQ logic reside in this module and they are running on the 33Mhz LPCCLK. Setting this bit will disable this module's ability to support CLKRUN# function from PCIBridge. In other words, when this bit is set, this module will prevent PCIBridge from stopping the 33Mhz clock
2	<b>test_pm_tmr.</b> Read-write. Reset: 0. <b>Description:</b> 0: normal PM timer function 1: PM timer 286riting286 run in test mode
1	<b>test_1ms.</b> Read-write. Reset: 0. <b>Description:</b> 0: normal delay in S state timing 1:Speed up S state delay by 7 times
0	<b>t32_64.</b> Read-write. Reset: 0. <b>Description:</b> Delay timer for S2: 0: 32 pci clk 1: 64 pci clk

**PMx000000BA (FCH::PM::s\_statecontrol)**

Read-write. Reset: 0000h.

\_aliasHOST; PMx000000BA; PM=FED8\_0300h

Bits	Description
15	<b>maskpmemsgen.</b> Read-write. Reset: 0. When set (along with PmeMsgEn=1), PmeAck message coming from PCIe® device will be ignored and ACPI S state logic will solely use the timeout mechanism to sequence through the S3 state. This bit is used as an option to guard against multiple PmeAck messages coming from CNB and internal FCH PCIe® bridge so FCH S state logic will not sequence into S3 state prematurely.
14	<b>wakepinenable.</b> Read-write. Reset: 0. Set to 1 to enable wakeup from WAKE# pin.
13	<b>agptimeadj.</b> Read-write. Reset: 0. If set to 1, S* -> S0 state transitions will use 1ms clock for timing sequence otherwise, 8s clock will be used.
12:6	Reserved.
5	<b>overrideshorttimemode.</b> Read-write. Reset: 0. <b>Description:</b> It is only available when ShortTimeMode is enabled. When this bit is set high, ShortTimeMode will be disabled for Osc16Us_new.
4	<b>pmemsgtrig.</b> Read-write. Reset: 0. <b>Description:</b> SW write this bit to trigger a PmeTurnOff sequence to NB. Reading this bit returns the status of the PmeTurnOff sequence (1 means not done 0 means done)
3	<b>pmemsgen.</b> Read-write. Reset: 0. Set to 1 to enable PmeTurnOff/PmeMsgAck handshake.
2	<b>stpcklen.</b> Read-write. Reset: 0. <b>Description:</b> 0: disable STPCLK/STPGNT handshake in S state transition 1: Enable STPCLK/STPGNT handshake in S state transition
1	<b>allowoffset.</b> Read-write. Reset: 0. Set to 1 to add extra delay for STPCLK. Only valid if AgpTimeAdj is set.
0	<b>longslps3.</b> Read-write. Reset: 0. Set to 1 to extend SLP_S3# assertion to 1s minimum.

**PMx000000BC (FCH::PM::espi\_vw\_control)**

Read-write. Reset: 0000h.

\_aliasHOST; PMx000000BC; PM=FED8\_0300h

Bits	Description
15	<b>rcinbrstenable.</b> Read-write. Reset: 0. Enable generating warm reset from eSPI RCIN# virtual wire
14	<b>hostrstwarndisable.</b> Read-write. Reset: 0. Enable generating early reset warning indicator and platform reset to eSPI
13:0	Reserved.

**PMx000000BE (FCH::PM::resetcontrol1)**

Read-write. Reset: 72h.

\_aliasHOST; PMx000000BE; PM=FED8\_0300h

Bits	Description
7	<b>rsttocpupwrgden.</b> Read-write. Reset: 0. If set to 1, FCH toggles CPUPG on every reset.
6	<b>hwm_resetooption.</b> Read-write. Reset: 1. <b>Description:</b> 0: Hwm function(Pmio2 register block) is reset by RsmRst. 1: Hwm function(Pmio2 register block) is reset by PciRst.
5	<b>slp_typen_control.</b> Read-write. Reset: 1. <b>Description:</b> Set to 1 to enable the function of SLP_TYPEn bit in PmControl register [AcpiPm1CntBlk:00h]. The SLP_TYPEn bit in PmControl register has no effect if this bit is clear.
4	<b>kbrsten.</b> Read-write. Reset: 1. Set to 1 to enable KB_RST# pin to trigger KB Reset.
3	<b>kb_trig_type.</b> Read-write. Reset: 0. <b>Description:</b> If kb_trig_type == 0, we will use the level trigger as kb_reset_src. If kb_sel_edge == 1, we will use edge trigger as kb_reset_src. This will give customers the choice how to trigger a keyboard reset.
2	<b>kb_sel_edge.</b> Read-write. Reset: 0. <b>Description:</b> This bit is applicable when kb_trig_type is set high. If kb_sel_edge == 0, negedge will be used as the kb_reset_src. If kb_sel_edge == 1, posedge will be used as the kb_reset_src.
1	<b>kb_pcirst_en.</b> Read-write. Reset: 1. Set to 1 to make PCI reset during KB Reset, which can be triggered by KB_RST# pin or EC.
0	<b>softreseten.</b> Read-write. Reset: 0. Set to 1 to block any reset request until the system is not C state.

**PMx000000BF (FCH::PM::resetcontrol2)**

Read-write. Reset: C0h.

\_aliasHOST; PMx000000BF; PM=FED8\_0300h

Bits	Description
7	<b>pwrgoodenb.</b> Read-write. Reset: 1. Output enable for PwrGood pin (active low)
6	<b>pwrgoodout.</b> Read-write. Reset: 1. Output data for PwrGood pin
5	<b>bypass_pwr_good.</b> Read-write. Reset: 0. If asserted, Southbridge will not wait for deassertion of PWRGOOD to monitor wakeup events.
4	<b>resetpcie2.</b> Read-write. Reset: 0. This bit is to be used with GEVENT4 mux select bits. If GEVENT4 IO mux select bits are set to 10b, GEVENT4# pin will become PCIE_RST2#
3	<b>shutdownpinen.</b> Read-write. Reset: 0. Shutdown system if seeing a negative edge on pinSHUTDOWN#
2	<b>overridewarmrst.</b> Read-write. Reset: 0. Add a S5 sticky bit "OverrideWarmRst" to override DevAllRstB so S5 GPIO can be persistent and not being reset by warm reset.
1	<b>selnew16us.</b> Read-write. Reset: 0. <b>Description:</b> When this bit is set high, Osc16Us_new will be used in PwrRst except MP1_WDT_cnt. When this bit is low, Osc16Us will be used in PwrRst.
0	Reserved.

**PMx000000C0 (FCH::PM::s5\_reset\_status)**

Reset: 0000\_0800h.

This register shows the source of previous reset.

\_aliasHOST; PMx000000C0; PM=FED8\_0300h

Bits	Description
31	<b>sw_sync_flood_flag.</b> Read-write,Read,Write-1-to-clear. Reset: 0. PMxC0[31] will be set if sw_sync_flood (PMx88[10]) trigger reset. Write 1 to clear. Bit[31] and Bit[28:16] except bit[20] will be cleared by Last reset event except the associated bit will be set.
30	<b>sdp_parity_err.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> When there is 'parity error', Sync_Flood reset will occur and PMxC0[27] will be set if enabled, in order to distinguish 'parity error' and 'CPU sync flood', PMxC0[30] will be set when there is SDP parity Error, thus software can distinguish 'parity error' and 'CPU sync flood'. This bit will not be cleared by other reset event, software need write 1 to clear. SDP parity error will not clear any status bit in this register.
29	<b>mp1_wdtout.</b> Read-write,Read,Write-1-to-clear. Reset: 0. This bit will be set to 1 when MP1_Watchdog timer time out (this indicates there was a failed warm reset handshake between SMU and FCH). This bit will not be cleared by other reset event, software need write 1 to clear. MP1_Watchdog timer time out will not clear any status bit in this register.
28	Reserved.
27	<b>sync_flood.</b> Read-write,Read,Write-1-to-clear. Reset: 0. system reset was caused by a SYNC_FLOOD event which was due to an UE error( when PMx74[18]=1). Write 1 to clear. Bit[31] and Bit[28:16] except bit[20] will be cleared by Last reset event except the associated bit will be set.
26	<b>remotereseetfromasf.</b> Read-write,Read,Write-1-to-clear. Reset: 0. system reset was caused by a remote RESET command from ASF. Write 1 to clear. Bit[31] and Bit[28:16] except bit[20] will be cleared by Last reset event except the associated bit will be set.
25	<b>watchdogissuereset.</b> Read-write,Read,Write-1-to-clear. Reset: 0. system reset was caused by WatchDog Timer. Write 1 to clear. Bit[31] and Bit[28:16] except bit[20] will be cleared by Last reset event except the associated bit will be set.
24	<b>failbootrst.</b> Read-write,Read,Write-1-to-clear. Reset: 0. system reset was caused by AMD Fail boot timer. Write 1 to clear. Bit[31] and Bit[28:16] except bit[20] will be cleared by Last reset event except the associated bit will be set.
23	<b>shutdown_msg.</b> Read-write,Read,Write-1-to-clear. Reset: 0. system reset was caused by a SHUTDOWN command from CPU (when PMx08[20]=1 and PMx74[17]=1). Write 1 to clear. Bit[31] and Bit[28:16] except bit[20] will be cleared by Last reset event except the associated bit will be set.
22	<b>kb_reset.</b> Read-write,Read,Write-1-to-clear. Reset: 0. system reset was caused by assertion of KB_RST_L. Write 1 to clear. Bit[31] and Bit[28:16] except bit[20] will be cleared by Last reset event except the associated bit will be set.
21	<b>sleepreset.</b> Read-write,Read,Write-1-to-clear. Reset: 0. Reset status from Sleep state (S0i3, S3, 4, or 5) transition. Write 1 to clear. Bit[31] and Bit[28:16] except bit[20] will be cleared by Last reset event except the associated bit will be set.
20	<b>do_k8_full_reset.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> system reset was caused by CF9 = 0x0E. Write 1 to clear. [Note] Write CF9=0xE will set this bit=1, but write CF9=0xE will generate SLpRst later which will set bit[21]=SleepReset. In order to keep this bit =1, this bit will not be cleared by hardware, software need to write 1 to clear this bit . Bit[31] and Bit[28:16] except bit[20] will be cleared by Last reset event except the associated bit will be set.
19	<b>do_k8_reset.</b> Read-write,Read,Write-1-to-clear. Reset: 0. system reset was caused by CF9 = 0x06. Write 1 to clear. Bit[31] and Bit[28:16] except bit[20] will be cleared by Last reset event except the associated bit will be set.
18	<b>do_k8_init.</b> Read-write,Read,Write-1-to-clear. Reset: 0. system reset was caused by CF9 = 0x04. Write 1 to clear. Bit[31] and Bit[28:16] except bit[20] will be cleared by Last reset event except the associated bit will be set.



17	<b>soft_pfirst.</b> Read-write,Read,Write-1-to-clear. Reset: 0. system reset was caused by writing to PMIO 0xC4[0] (PciReset). Write 1 to clear. Bit[31] and Bit[28:16] except bit[20] will be cleared by Last reset event except the associated bit will be set.
16	<b>userrst.</b> Read-write,Read,Write-1-to-clear. Reset: 0. Last reset was caused by BP_SYS_RST_L assertion. Write 1 to clear. Bit[31] and Bit[28:16] except bit[20] will be cleared by Last reset event except the associated bit will be set.
15:14	<b>pmeturnofftime.</b> Read-write. Reset: 0h. <b>Description:</b> 00: 1ms 01: 2ms 10: 4ms 11: 8ms
13:10	Reserved.
9	<b>intthermaltrip.</b> Read-write,Read,Write-1-to-clear. Reset: 0. system was shut down due to an internal ThermalTrip event. Write 1 to clear
8:5	Reserved.
4	<b>remotepowerdownfromasf.</b> Read-write,Read,Write-1-to-clear. Reset: 0. SOC has received a remote Power Off command from ASF. Write 1 to clear.
3	Reserved.
2	<b>shutdown.</b> Read-write,Read,Write-1-to-clear. Reset: 0. system was shut down due to ShutDown event (SHUTDOWN# pin). Write 1 to clear.
1	<b>pwrbtn4second.</b> Read-write,Read,Write-1-to-clear. Reset: 0. system was shut down due to 4s PwrButton event. Write 1 to clear.
0	<b>thermaltrip.</b> Read-write,Read,Write-1-to-clear. Reset: 0. system was shut down due to BP_THERMTRIP_L assertion. Write 1 to clear.

**PMx000000C4 (FCH::PM::resetcommand)**

Read-write. Reset: 28h.

\_aliasHOST; PMx000000C4; PM=FED8\_0300h

Bits	Description
7	<b>reseten.</b> Read-write. Reset: 0. <b>Description:</b> 0: Not allow to write bit 0 1: Allow to write bit 0.
6	<b>resetallacpi.</b> Read-write. Reset: 0. Writing 1 to emulate a Reset Button event.
5	<b>resetbuttonen.</b> Read-write. Reset: 1. <b>Description:</b> 1: reset from reset button (SYS_RESET_L) will generate S5Reset. 0: Disable SYS_RESET_L to generate S5Reset, it can be used as Gevent19/GPIO1.
4	<b>resetpcie.</b> Read-write. Reset: 0. Set to 1 to reset Gpp port.
3	<b>usrrst2pll.</b> Read-write. Reset: 1. Set to 1 to stop Pll when reset button is pressed.
2	<b>selectdebug.</b> Read-write. Reset: 0. <b>Description:</b> 0: Select the PM_Reg C0 to be S5/Reset Status register. 1: Select the PM_Reg C0 to be a debug status register.
1	<b>memrstdisable.</b> Read-write. Reset: 0. When set, the memory reset function at DDR_RST# pin will be disabled.
0	<b>pcireset.</b> Read-write. Reset: 0. Writing 1 to do a PCI reset



**PMx000000C5 (FCH::PM::cf9shadow)**

Read-write. Reset: 00h.

\_aliasHOST; PMx000000C5; PM=FED8\_0300h

Bits	Description
7:4	Reserved.
3	<b>fullrst.</b> Read-write. Reset: 0. <b>Description:</b> 0: Assert reset signals only 1: Place system in S5 state for 3 to 5 seconds
2	<b>rstcmd.</b> Read-write. Reset: 0. <b>Description:</b> Write with 1 to generate reset as specified by bit[3,1]. Write only. Always read as 0.
1	<b>sysrst.</b> Read-write. Reset: 0. <b>Description:</b> 0: Send INIT HT message 1: Reset as specified by bit3
0	Reserved.

**PMx000000C6 (FCH::PM::htcontrol)**

Read-write. Reset: 0000h.

\_aliasHOST; PMx000000C6; PM=FED8\_0300h

Bits	Description
15:0	Reserved.

**PMx000000C8 (FCH::PM::misc\_pmio)**

Read-write. Reset: 0028\_800Ch.

\_aliasHOST; PMx000000C8; PM=FED8\_0300h

Bits	Description
31:24	<b>clkintrvectorord.</b> Read-write. Reset: 00h. Specify the value used to indentify the clock interrupt.
23	Reserved.
22	<b>clkintrvectororden.</b> Read-write. Reset: 0. When set, the system timer interrupt in the IOAPIC will be tagged with a value defined by ClkIntrVectorOrd
21	<b>align_s3s5.</b> Read-write. Reset: 1. <b>Description:</b> 0: Old scheme, BP_SLP_S5_L and BP_SLP_S3_L may not be aligned, this is default. 1: new scheme, BP_SLP_S5_L and BP_SLP_S3_L are aligned.
20	Reserved.
19	<b>usecpurst.</b> Read-write. Reset: 1. If this bit is not set, system reset will cause INIT# instead of CPURST#.
18:16	Reserved.
15	<b>rsmu_cpl_gate_dis.</b> Read-write. Reset: 1. <b>Description:</b> Chicken bit to disable the gating for Cpl signals between RSMU and SMU. If it is low, gating enabled. If it is high, gating disabled.
14	<b>ocpupwrgood_sel.</b> Read-write. Reset: 0. <b>Description:</b> Chicken bit to disable the gating for signals between RSMU and SMU. If it is low, gating enabled. If it is high, gating disabled.
13	<b>id_change_en.</b> Read-write. Reset: 0. Setting this bit will allow the software to change the DeviceID and RevisionID.
12	<b>s5resetoverride.</b> Read-write. Reset: 0. Set to 1 to mask off internet PCI reset used in ACPI.
11	<b>writebackenable.</b> Read-write. Reset: 0. HD audio/modem write back enable. If set, the WakeOnRing status bit will be written back to HD Audio controller upon system power up.
10	<b>llb_en.</b> Read-write. Reset: 0. If set, LLB function is enabled, and system won't wakeup from ACPI S state until LLB# is de-asserted.
9:8	<b>temp_polarity.</b> Read-write. Reset: 0h. <b>Description:</b> Temperature polarity control for THRMTRIP and TALERTRIP respectively. 0: Active low 1: Active high
7	Reserved.
6	<b>twarnen.</b> Read-write. Reset: 0. If set, it enables TALERTRIP pin
5:4	Reserved.
3	<b>tdeaden.</b> Read-write. Reset: 1. When set, GEVENT2 takes up the THRMTRIP function. When THRMTRIP pin is low and TFATAL_EN(bit2 of the same register) is set, hardware will switch the system to S5 automatically.
2	<b>tfatal_en.</b> Read-write. Reset: 1. This bit enables both the soft PCIRST and the THRMTRIP function.
1	<b>instatnoffenable.</b> Read-write. Reset: 0. Enable fast shutdown upon THERMTRIP# event
0	<b>cpu_io_pulldowndrvstrength.</b> Read-write. Reset: 0. When set, the integrated pull-down drive strength of all CPU Ios are increased by 50%.

**PMx000000CC (FCH::PM::iodrvsth)**

Read-write. Reset: DB6D\_B6DBh.

\_aliasHOST; PMx000000CC; PM=FED8\_0300h

Bits	Description
31:0	Reserved.

**PMx000000D2 (FCH::PM::pmiodebug)**

Read-write. Reset: 30h.

\_aliasHOST; PMx000000D2; PM=FED8\_0300h

Bits	Description
7	Reserved.
6	<b>cf9rstdisable</b> . Read-write. Reset: 0. When set, write to CF9 will not generate a reset. The purpose of this bit is to allow BIOS to trap CF9
5:0	Reserved.

**PMx000000D4 (FCH::PM::pwrrstdebsel)**

Read-write. Reset: 00h.

\_aliasHOST; PMx000000D4; PM=FED8\_0300h

Bits	Description
7:5	Reserved.
4:0	<b>debugseldev</b> . Read-write. Reset: 00h. Debug Bus Select for PwrRst signals

**PMx000000D5 (FCH::PM::altmmioen)**

Read-write. Reset: 00h.

\_aliasHOST; PMx000000D5; PM=FED8\_0300h

Bits	Description
7:3	Reserved.
2	<b>lock_wr</b> . Read-write. Reset: 0. <b>Description:</b> 0: PMxD4[31:8] can be written. 1: PMxD4[31:8] cannot be written
1	<b>alt_addr_width_sel</b> . Read-write. Reset: 0. <b>Description:</b> 1h : alternate address is 64bit width, 0h : alternate address is 32bit width. Value will be locked when PMxD5[2] =1
0	<b>alt_addr_en</b> . Read-write. Reset: 0. <b>Description:</b> Alternate address enable bit. Value will be locked when PMxD5[2] =1

**PMx000000D6 (FCH::PM::altmmiobase)**

Read-write. Reset: 0000h.

\_aliasHOST; PMx000000D6; PM=FED8\_0300h

Bits	Description
15:0	<b>lower_addr_alt</b> . Read-write. Reset: 0000h. <b>Description:</b> Lower bits of base address. Its value only take effect when alternate address is enabled. For example, when it is set as 0xABCD and alternate address width is 32bit, base address will be 0xABCD_0000. If it is 64 bit then the base address is 0xFFFF_FFFF_ABCD_0000. Value will be locked when PMxD5[2] =1

**PMx000000D8 (FCH::PM::eprom\_efuseindex)**

Read-write. Reset: 00h.

\_aliasHOST; PMx000000D8; PM=FED8\_0300h

Bits	Description
7:0	Reserved.

**PMx000000D9 (FCH::PM::eprom\_efusedata)**

Read-write. Reset: 00h.

\_aliasHOST; PMx000000D9; PM=FED8\_0300h

**Bits Description**

7:0 Reserved.

**PMx000000DA (FCH::PM::sataconfig)**

Read-write. Reset: 0011h.

\_aliasHOST; PMx000000DA; PM=FED8\_0300h

**Bits Description**15:8 **pmio\_xda\_sataconfig**. Read-write. Reset: 00h. Not used.7:6 **ref\_div\_sel**. Read-write. Reset: 0h.**Description:** This is CP\_PLL\_CLKR, the reference clock divider setting.

00: Divide by 1 (25MHz reference clock)

01: Divide by 2

10: Divide by 4 (100MHz reference clock)

11: Same as 10

5:4 **ref\_clk\_sel**. Read-write. Reset: 1h.**Description:** This is CP\_PLL\_REFCLK\_SEL, the reference clock source selection for SATA PLL.

00/10: Reference clock from crystal oscillator via PAD\_XTALI and PAD\_XTALO

01: Reference clock from internal clock through CP\_PLL\_REFCLK\_P and CP\_PLL\_REFCLK\_N via RDL

11: same as 01

3 **hiddenide**. Read-write. Reset: 0.**Description:** 0: IDE controller is exposed and Combined Mode is enabled. SATA controller has control over Port0 through Port3, IDE controller has control over Port4 and Port5

1: IDE controller is hidden and Combined Mode is disabled, SATA controller has full control of all 6 Ports when operating in non-IDE mode

2 **setmaxgen2**. Read-write. Reset: 0.**Description:** 0: SATA controller operates in maximum Gen3 (3.0Gbps) speed

1: SATA controller operates in maximum Gen2 (3.0Gbps) speed and saves more power on PLL.

1 **channel\_sel**. Read-write. Reset: 0.**Description:** 0: SATA Port4 and Port5 utilizing Primary IDE channel

1: SATA Port4 and Port5 utilizing Secondary IDE channel

0 **sataenable**. Read-write. Reset: 1.**Description:** 0: SATA controller is disabled

1: SATA controller is enabled

**PMx000000DC (FCH::PM::pmio\_xdc\_sataconfig)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; PMx000000DC; PM=FED8\_0300h

**Bits Description**31:0 **pmio\_xdc\_sataconfig**. Read-write. Reset: 0000\_0000h.**Description:** Used by Sata DEVSLP.

This register only reset by RsmRstB or UserRstB, BIOS need to clear this register to 0x0 after WarmReset so that device can response with SATA controller's OOB.

**PMx000000E0 (FCH::PM::abregbar)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; PMx000000E0; PM=FED8\_0300h

Bits	Description
31:16	Reserved.
15:3	<b>abregbar</b> . Read-write. Reset: 0000h. The 13 bits are mapped into AB IO base[15:3]. AB IO Base [2:0] are tied to 0.
2:0	Reserved.

**PMx000000E4 (FCH::PM::abdebug)**

Read-write. Reset: 0000h.

\_aliasHOST; PMx000000E4; PM=FED8\_0300h

Bits	Description
15	<b>ab_jtagmuxhost_en</b> . Read-write. Reset: 0. oAB_JtagMuxHost_en
14:7	Reserved.
6	<b>sdpdebugmode</b> . Read-write. Reset: 0. Select the SDP Debug mode.
5	<b>sdpdebugen</b> . Read-write. Reset: 0. Enable the SDP Debug mode.
4	<b>reqfilteron</b> . Read-write. Reset: 0. Not used.
3:2	<b>arbcontrol</b> . Read-write. Reset: 0h. Not used.
1:0	<b>blinkcontrol</b> . Read-write. Reset: 0h. <b>Description:</b> Control the behavior on Blink pad. // 00 - off // 01 - 1/4 hz, // 10 - 1/2 hz, // 11 - always on

**PMx000000E6 (FCH::PM::dacntrl)**

Read-write. Reset: 0001h.

\_aliasHOST; PMx000000E6; PM=FED8\_0300h

Bits	Description
15:0	<b>dacntrl</b> . Read-write. Reset: 0001h. For VGA tile

**PMx000000E8 (FCH::PM::sdflashcntrl)**

Read-write.

\_aliasHOST; PMx000000E8; PM=FED8\_0300h

Bits	Description
7:6	Reserved.
5	<b>sd_cd2acpi</b> . Read-write. Reset: 0. Not used.
4	<b>emmc_en</b> . Read-write. Reset: 0. Not used.
3	<b>sdio_en</b> . Read-write. Reset: 0. Not used.
2	<b>sd_mem_dsd_bypass</b> . Read-write. Reset: 1. Not used.
1	<b>sdtest</b> . Read-write. Reset: 0. Not used.
0	<b>sdflashenable</b> . Read-write. Reset: 0. <b>Description:</b> 0: Disable SD flash controller 1: Enable SD flash controller Whenever this bit is set, GPIO[73:80] will become SD flash interface Not used.

**PMx000000EA (FCH::PM::ufscntrl)**

Read-write. Reset: 00h.

\_aliasHOST; PMx000000EA; PM=FED8\_0300h

Bits	Description
7:4	Reserved.
3	<b>mphy_pwrokvddc.</b> Read-write. Reset: 0. <b>Description:</b> 0: No deassertion or don't force PwrOkVddc going to MPHY to low 1: Deassert or force PwrOkVddc going to MPHY to low
2	<b>mphy_cpbypassrefclk_en.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pad is output or Enable 26MHz output or Disable BypRefClk input 1: Pad is input or Disable 26MHz output or Enable BypRefClk input
1	<b>byp_mphy_debugbus.</b> Read-write. Reset: 0. <b>Description:</b> 0: Don't Bypass MPHY DSM Debug Bus 1: Bypass MPHY DSM Debug Bus
0	<b>dis26mhzpadgt.</b> Read-write. Reset: 0. <b>Description:</b> 0: Enable 26MHz pad gating as per design logic 1: Disable 26MHz pad gating and always enable 26 MHz output

**PMx000000EB (FCH::PM::azen)**

Read-write. Reset: 01h.

\_aliasHOST; PMx000000EB; PM=FED8\_0300h

Bits	Description
7:2	Reserved.
1	<b>aznosnoopenable.</b> Read-write. Reset: 0. <b>Description:</b> When set, HD AUDIO data transfer will not cause the BM_STS bit to be set and to wake up the CPU from C3 state. Under current C1e implementation, there is no need to set this bit.
0	<b>azenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Disable HD audio controller 1: Enable HD audio controller

**PMx000000EC (FCH::PM::lpcgating)**

Read-write. Reset: 11h.

\_aliasHOST; PMx000000EC; PM=FED8\_0300h

Bits	Description
7:6	Reserved.
5	<b>iss5rom.</b> Read-write. Reset: 0. <b>Description:</b> 1: Rom_Rst output is deasserted from G3 to S0. 0: Rom_Rst output is toggled along with Arst#.
4	<b>dislpctoacpi.</b> Read-write. Reset: 1. <b>Description:</b> 1: The ReqB from ACPI won't be blocked when there is cycle to LPC. 0: The ReqB from ACPI will be blocked when there is cycle to LPC. (This bit is also related to bit 3.)
3	<b>pendinglpcdmaiocyclefix.</b> Read-write. Reset: 0. Enable the fix for #SB02634
2	<b>abnobypassen.</b> Read-write. Reset: 0. Set to 0 to not allow AB bypass any LPC cycles.
1	<b>lpc_a20en.</b> Read-write. Reset: 0. Set to 1 to enable A20# input.
0	<b>lpc_enable.</b> Read-write. Reset: 1. Set to 1 to enable LPC bridge

**PMx000000ED (FCH::PM::usb gating)**

Read-write. Reset: 00h.

\_aliasHOST; PMx000000ED; PM=FED8\_0300h

Bits	Description
7	Reserved.
6	<b>usb_smiact_en.</b> Read-write. Reset: 0. Not used.
5	<b>undo_smi_change.</b> Read-write. Reset: 0. Not used.
4	<b>usb_smi_en.</b> Read-write. Reset: 0. Set to 1 to enable USB SMI#.
3	<b>usb_pme_gate.</b> Read-write. Reset: 0. Set this bit with 1 to block USB3_HC0/1, USBADP_USB3_0/1/2, USB4_0/1/2 PME.
2	<b>usb_drd_en.</b> Read-write. Reset: 0. Set this bit with 1 to block USBADP_USB3_0/1/2 DRD interrupts
1	<b>usb_irq_en.</b> Read-write. Reset: 0. Set to 1 to route IRQ1/IRQ12 from usb to PIC/IOAPIC.
0	<b>usb_a20_en.</b> Read-write. Reset: 0. Set to 1 to enable USB A20#.

**PMx000000EE (FCH::PM::usb3cntrl)**

\_aliasHOST; PMx000000EE; PM=FED8\_0300h

Bits	Description
7:6	Reserved.
5:4	<b>xhc1powersel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: oXhc1S5RstB assert when CF9, KbRst, RsmRst, UsrRst, S5, S3, S0i3. 01: oXhc1S5RstB assert when CF9, KbRst, RsmRst, UsrRst, S5. 10: oXhc1S5RstB assert when CF9, KbRst, RsmRst, UsrRst. 11: oXhc1S5RstB assert when CF9, KbRst, RsmRst, UsrRst.
3	<b>usb3efusstat.</b> Read-only. Reset: X. This bit means "USB3 is Efuse-enabled"
2	Reserved.
1:0	<b>xhc0powersel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: oXhc0S5RstB assert when CF9, KbRst, RsmRst, UsrRst, S5, S3, S0i3. 01: oXhc0S5RstB assert when CF9, KbRst, RsmRst, UsrRst, S5. 10: oXhc0S5RstB assert when CF9, KbRst, RsmRst, UsrRst. 11: oXhc0S5RstB assert when CF9, KbRst, RsmRst, UsrRst.

**PMx000000EF (FCH::PM::usbdebug)**

Read-write. Reset: 7Fh.

\_aliasHOST; PMx000000EF; PM=FED8\_0300h

Bits	Description
7:0	<b>s3_usbenable.</b> Read-write. Reset: 7Fh. Not used

**PMx000000F0 (FCH::PM::usbcontrol)**

Read-write. Reset: 739Ch.

\_aliasHOST; PMx000000F0; PM=FED8\_0300h

Bits	Description
15:13	Reserved.
12	<b>usb2blglobalclkgateen.</b> Read-write. Reset: 1. Set to 1 to enable USB2 B-Link Global Clock Gating
11	Reserved.
10:8	<b>usbsleepctrl.</b> Read-write. Reset: 3h. <b>Description:</b> Control on USB advanced async sleep function. Setting of 000b:100b are for the advanced async sleep. 000b: Standard 10s sleep 001b: Advanced sleep up to 2 uframes 010b: Advanced sleep up to 4 uframes 011b: Advanced sleep up to 6 uframes (default) 100b: Advanced sleep up to next uframe 0 101b, 110b: Reserved 111b: EHCI will stop fetching descriptor once it has completed the list while CPU is in C state.
7	<b>usbresumeenable.</b> Read-write. Reset: 1. Set to 1 to enable S3 wakeup on USB device resume.
6	<b>usbs3wakeresumeonlydisable.</b> Read-write. Reset: 0. Default (0) is to support USB Wake-Up event on Resume only. When set to 1, USB can wakeup on all wake events, connection, disconnect, over-current, and/or resume detect.
5	<b>pmio_ohci_mem_slp_dis.</b> Read-write. Reset: 0. <b>Description:</b> 0: enable OHCI memory sleep mode 1: disable OHCI memory sleep mode
4	<b>usb11pdresistorenable.</b> Read-write. Reset: 1. Set to 0 to disconnect pull-down resistors on stand-alone USB1.1 pads.
3	<b>usbs5resetenable.</b> Read-write. Reset: 1. Set to 1 to enable USB reset on S4/S5 resume detection.
2	<b>usbkbreasetenable.</b> Read-write. Reset: 1. Set to 1 to enable resetting USB on KB reset.
1	<b>usbforceregenable.</b> Read-write. Reset: 0. This function is not implemented.
0	<b>usbphys5pwrdownenable.</b> Read-write. Reset: 0. <b>Description:</b> Set to 1 to disable S4/S5 USB PHY power down support and to enable S4 USB wakeup support. The bit has to be set to 1 to support S4 USB wakeup.

**PMx000000F2 (FCH::PM::usbcontrol\_1)**

Read-write. Reset: 00h.

\_aliasHOST; PMx000000F2; PM=FED8\_0300h

Bits	Description
7:0	Reserved.

**PMx000000F3 (FCH::PM::usbdebug\_2)**

Read-write. Reset: 00h.

\_aliasHOST; PMx000000F3; PM=FED8\_0300h

Bits	Description
7	<b>forcephyportreset.</b> Read-write. Reset: 0. Forces USB PHY port reset.
6	<b>forcephyearlyreset.</b> Read-write. Reset: 0. Forces USB PHY early reset.
5	<b>forcephydllreset.</b> Read-write. Reset: 0. Forces USB PHY DLL reset.
4	<b>forcephypllreset.</b> Read-write. Reset: 0. Forces USB PHY PLL reset.
3	<b>forcephyprdown.</b> Read-write. Reset: 0. Forces USB PHY into power down mode.
2:0	<b>forcereset2usb.</b> Read-write. Reset: 0h. These are software control bits that can be used to force resetting of USB host controllers. Each bit corresponds to one USB major function.



**PMx000000F4 (FCH::PM::usbcontrol\_2)**

Read-write. Reset: 0000h.

\_aliasHOST; PMx000000F4; PM=FED8\_0300h

Bits	Description
15:3	Reserved.
2	<b>usb_s3_ohci_discon_fix.</b> Read-write. Reset: 0. Not used.
1	<b>ohci_hidden_enable.</b> Read-write. Reset: 0. <b>Description:</b> This bit is used to hide OHCI controllers inside USB2.0. 0: USB2.0 contains OHCI and EHCIs. 1: All OHCI controllers inside USB2.0 are hidden and only EHCI controller exist as single function. Not used.
0	<b>usb_s3_dis_con_wo_wakeen.</b> Read-write. Reset: 0. <b>Description:</b> Set to open EHCI/OHCI S3 disconnect and connect fix. Not used.

**PMx000000F6 (FCH::PM::otgcfg)**

Read-write. Reset: 0000h.

\_aliasHOST; PMx000000F6; PM=FED8\_0300h

Bits	Description
15:2	Reserved.
1:0	<b>otgpowersel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: oOtgS5RstB assert when CF9, KbRst, RsmRst, UsrRst, S5, S3, S0i3. 01: oOtgS5RstB assert when CF9, KbRst, RsmRst, UsrRst, S5. 10: oOtgS5RstB assert when CF9, KbRst, RsmRst, UsrRst. 11: oOtgS5RstB assert when CF9, KbRst, RsmRst, UsrRst.

**PMx000000F8 (FCH::PM::spbbase)**

Read-write. Reset: FEF0\_003Fh.

\_aliasHOST; PMx000000F8; PM=FED8\_0300h

Bits	Description
31:0	<b>spbs5cntrl.</b> Read-write. Reset: FEF0_003Fh. Not used.

**PMx000000FC (FCH::PM::TRACEMEMORYEN)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; PMx000000FC; PM=FED8\_0300h

Bits	Description
31:20	<b>tracememorybaseaddr.</b> Read-write. Reset: 000h. The base address of trace memory in ILA (Internal Logic Analyzer) connect to ILA.ILA_Mem_Addr. It is 1M memory space.
19:1	Reserved.
0	<b>tracememoryen.</b> Read-write. Reset: 0. Set to 1 to enable trace memory decoding in ILA (Internal Logic Analyzer), connect to ILA.Acpi2ILA_Enable

**PMx00000500 (FCH::PM::pm1status)**

This register is located at the base address defined by AcpiPm1EvtBlk.

\_aliasHOST; PMx00000500; PM=FED8\_0300h

Bits	Description
15	<b>wakestatus.</b> Read-write. Reset: 0. This bit is set when the system is in the sleep state and a wake-up event occurs.
14	<b>pciexpwakestatus.</b> Read-write. Reset: 0. This bit is set by hardware to indicate that the system wake is due to a PCI Express® wakeup event.
13:11	Reserved.
10	<b>rtcstatus.</b> Read-write. Reset: 0. This bit is set when RTC generates an alarm.
9	Reserved.
8	<b>pwrbtnstatus.</b> Read-write. Reset: 0. Power button status bit
7:6	Reserved.
5	<b>gblstatus.</b> Read-write. Reset: 0. This bit is set when an SCI is generated due to the BIOS wanting the attention of the SCI handler. This is set by writing 1 to PM_Reg: 74h bit [7].
4	<b>bmstatus.</b> Read-write. Reset: 0. Bus master status bit. This bit is set any time a system bus master requests the system bus, and can only be cleared by writing an one to this bit position. Note: this bit is no longer used except for server CPU. For server CPU, this will be used in conjunction with IDLE_EXIT# pin
3:1	Reserved.
0	<b>tmrstatus.</b> Read-only. Reset: 0. Timer carry status bit. This bit gets set anytime the 31st bit of 32 bit counter changes (whenever the MSB changes from low to high or high to low. While TmrEn and TmrStatus are set, an interrupt event is raised).

**PMx00000502 (FCH::PM::pm1enable)**

Read-write.

This register is located at the base address defined by AcpiPm1EvtBlk.

\_aliasHOST; PMx00000502; PM=FED8\_0300h

Bits	Description
15	Reserved.
14	<b>pciexpwakedis.</b> Read-write. Reset: 1. This bit disables the inputs to the PciExpWakeStatus from waking the system.
13:11	Reserved.
10	<b>rtcen.</b> Read-write. Reset: 0. RTC enable. If this bit is set, SCI is generated whenever RtcStatus is true.
9	Reserved.
8	<b>pwrbtnen.</b> Read-write. Reset: 0. If this bit is set, SCI is generated whenever PwrBtnStatus is true.
7:6	Reserved.
5	<b>gblen.</b> Read-write. Reset: 0. If this bit is set, SCI is raised whenever both GblEn and GblStatus are true.
4:1	Reserved.
0	<b>tmren.</b> Read-write. Reset: 0. This is the timer carry interrupt enable bit. When this bit is set then an SCI event is generated anytime the TmrStatus is set. When this bit is reset then no interrupt is generated when the TmrStatus bit is set.

**PMx00000504 (FCH::PM::pmcontrol)**

Read-write.

This register is located at the base address defined by AcpiPm1CntBlk (PM\_Reg:62h).

\_aliasHOST; PMx00000504; PM=FED8\_0300h

Bits	Description
15:14	Reserved.
13	<b>slp_en.</b> Read-write. Reset: 0. <b>Description:</b> This is a write-only bit and reads from it always return zero. If PM_Reg:Beh bit5 (SLP_TYP_EN) is 1 (default), setting this bit will cause the system to sequence into the sleeping state associated with the SLP_TYP fields programmed. Writing 0 to this bit has no effect.
12:10	<b>slp_typ.</b> Read-write. Reset: 0h. Defines the sleep state the system enters when the SLP_TYPEn is set to one. This design currently implements 5 states: S0, S1, S3, S4, and S5.
9:3	Reserved.
2	<b>gbl_rls.</b> Read-write. Reset: 0. If SMI_Reg:B0h[17:16] is set to 01b, writing 1 to this bit will generate SMI# and set SMI_Reg:88h bit[8]. Reading, this bit will always return 0.
1	<b>bmrlld.</b> Read-write. Reset: 0. If this bit is set, any bus master activity will cause the C state logic to break out from C3. This is no longer needed for current C state implementation
0	<b>sci_en.</b> Read-write. Reset: 0. Selects the power management event to be either an SCI or SMI# interrupt for the following events. When this bit is set, then PM events will generate an SCI interrupt otherwise, it will be SMI#.

**PMx00000508 (FCH::PM::tmrvalue\_etmrvalue)**

Read-only.

AcpiPmTmrBlk is defined in PM\_Reg:64h

\_aliasHOST; PMx00000508; PM=FED8\_0300h

Bits	Description
31:0	<b>tmrvalue.</b> Read-only. Reset: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXb. This read-only field returns the running count of the power management timer (ACPI timer).

**PMx0000050C (FCH::PM::clkvalue)**

Read-write. Reset: 0000\_0000h.

CpuControl Io base is defined in PM\_Reg 0x66

\_aliasHOST; PMx0000050C; PM=FED8\_0300h

Bits	Description
31:0	Reserved.

**PMx00000510 (FCH::PM::plvl2)**

Read-only. Reset: 00h.

CpuControl Io base is defined in PM\_Reg 0x66

\_aliasHOST; PMx00000510; PM=FED8\_0300h

Bits	Description
7:0	<b>plvl2.</b> Read-only. Reset: 00h. Reads to this register return all zeros and generate a "enter C2" sequence to APU writes to this register have no effect.

**PMx00000511 (FCH::PM::plvl3)**

Read-only. Reset: 00h.

CpuControl Io base is defined in PM\_Reg 0x66

\_aliasHOST; PMx00000511; PM=FED8\_0300h

Bits	Description
7:0	<b>plvl3.</b> Read-only. Reset: 00h. Reads to this register return all zeros and generate a "enter C3" sequence to APU writes to this register have no effect.

**PMx00000514 (FCH::PM::event\_status\_stdacpi)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; PMx00000514; PM=FED8\_0300h

Bits	Description
31:0	<b>eventstatus.</b> Read-write. Reset: 0000_0000h. Each bit represents an ACPI event status. Writing 1 to any bit clears it. Each event status is set when the selected event input equals to the corresponding value in SciTrig. Note: configuration for EVENTS are located at SMI_Reg: 08h through 70h. The status bits are also mirrored in SMI_Reg: 00h

**PMx00000518 (FCH::PM::event\_enable\_stdacpi)**

Read-write. Reset: 0000\_0000h.

\_aliasHOST; PMx00000518; PM=FED8\_0300h

Bits	Description
31:0	<b>eventenable.</b> Read-write. Reset: 0000_0000h. Each bit controls whether ACP should generate wakeup and Sci interrupt. The enable bits are also mirrored in SMI_Reg: 04h

**PMx0000051C (FCH::PM::smicmdport)**

Read-write. Reset: 00h.

This register is located at the base address defined by AcpiSmiCmd [PM\_Reg:6Ah] + offset 0. When SMI command port is enabled, write to this port will generate SMI# (only IOW can generate SMI#, MEMW will not generate SMI#). A read of this address will return the previously written value but will not generate SMI. The SMI command port has to be located at an even address (ie, 0, 2, 4, 6, 8, A, C, or E). SmiCmdStatus is always located one byte immediately after the SmiCmdPort. The implementation actually contains four (4) bytes of address space. If SMI command port is assigned to byte 0, then byte 1 is SmiCmdStatus. Byte 2 and 3 can be used as scratch. If SmiCmdPort is assigned to byte 2, then bytes 0 and 1 are not available.

\_aliasHOST; PMx0000051C; PM=FED8\_0300h

Bits	Description
7:0	<b>smicmdport.</b> Read-write. Reset: 00h. Writing the Port can generate Smi.

**PMx0000051D (FCH::PM::smicmdstatus)**

Read-write. Reset: 00h.

This register is located at the base address defined by AcpiSmiCmd [PM\_Reg:6Ah] + offset 1.

\_aliasHOST; PMx0000051D; PM=FED8\_0300h

Bits	Description
7:0	<b>smicmdstatus.</b> Read-write. Reset: 00h. Used by BIOS and OS

**PMx00000524 (FCH::PM::pmacontrol)**

Read-write.

AcpiPmaCntBlk is defined in PM\_Reg:6Eh

\_aliasHOST; PMx00000524; PM=FED8\_0300h

Bits	Description
7:1	Reserved.
0	<b>arb_dis.</b> Read-write. Reset: 0. System arbiter is disabled when this bit is set. Note: under this current AMD C state implementation, this is no longer used and should not be reported to OS.

**PMx00000528 (FCH::PM::pmacontrol\_1)**

Read-write.

AcpiPmaCntBlk is defined in PM\_Reg:6Eh

\_aliasHOST; PMx00000528; PM=FED8\_0300h

Bits	Description
7:1	Reserved.
0	<b>arb_dis2.</b> Read-write. Reset: 1. System arbiter is disabled when this bit is set if ARB_DIS is also set. This bit is hidden from OS and meant to be used by BIOS.

## 7.3.9.3 RTC External Registers

**PMx5F\_x00 [RTCEXT DltSavEn] (FCH::PM::RTCEXT::RTCEXTDltSavEn)**

Read-write.

\_aliasHOSTLEGACY; PMx5F\_x00; PM=FED8\_0300h; DataPortWrite=FCH::PM::rtcextindex

Bits	Description
7:1	Reserved.
0	<b>DltSavEnable</b> . Read-write. Reset: X. 1=Enable RTC daylight saving feature. The value of this register is undefined/non-deterministic when powered up for the first time.

**PMx5F\_x01 [RTCEXT SprFwdCtrl] (FCH::PM::RTCEXT::RTCEXTSprFwdCtrl)**

Read-write.

\_aliasHOSTLEGACY; PMx5F\_x01; PM=FED8\_0300h; DataPortWrite=FCH::PM::rtcextindex

Bits	Description										
7	Reserved.										
6	<b>SprFwdWeek</b> . Read-write. Reset: X. This specifies which Sunday morning to do the "spring forward". Spring forward is usually at the 1st Sunday of April in United States and last Sunday of March in Europe. 0=The 1st Sunday of the month. 1=The last Sunday of the month. The value of this register is undefined/non-deterministic when powered up for the first time.										
5:0	<b>SprFwdHour</b> . Read-write. Reset: XXXXXXb. <b>Description:</b> This Binary-Coded Decimal (BCD) value determines which hour (24 hour mode) to do the "spring forward". Setting of 02h means 2am. Default is 00h which also denotes 2am. Spring forward is usually 2am in United States and 1am in Europe. Note: the value of this register is undefined/indeterministic when power up first time. <b>ValidValues:</b> <table border="1"> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>00h</td><td>2AM.</td></tr> <tr> <td>01h</td><td>1AM.</td></tr> <tr> <td>02h</td><td>2AM.</td></tr> <tr> <td>3Fh-03h</td><td>Reserved.</td></tr> </tbody> </table>	Value	Description	00h	2AM.	01h	1AM.	02h	2AM.	3Fh-03h	Reserved.
Value	Description										
00h	2AM.										
01h	1AM.										
02h	2AM.										
3Fh-03h	Reserved.										

**PMx5F\_x02 [RTCEXT SprFwdMonth] (FCH::PM::RTCEXT::RTCEXTSprFwdMonth)**

Read-write.

\_aliasHOSTLEGACY; PMx5F\_x02; PM=FED8\_0300h; DataPortWrite=FCH::PM::rtcextindex

Bits	Description						
7:5	Reserved.						
4:0	<b>SprFwdMonth</b> . Read-write. Reset: XXXXXb. This Binary-Coded Decimal (BCD) value determines which month to do the "spring forward". The value of this register is undefined/non-deterministic when powered up for the first time. <b>ValidValues:</b> <table border="1"> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>0</td><td>April</td></tr> <tr> <td>1</td><td>March</td></tr> </tbody> </table>	Value	Description	0	April	1	March
Value	Description						
0	April						
1	March						

**PMx5F\_x03 [RTCEXT FallBackCtrl] (FCH::PM::RTCEXT::RTCEXTFallBackCtrl)**

Read-write.

\_aliasHOSTLEGACY; PMx5F\_x03; PM=FED8\_0300h; DataPortWrite=FCH::PM::rtcextindex

Bits	Description																																																										
7	Reserved.																																																										
6	<b>FallBackWeek.</b> Read-write. Reset: X. 0=The last week of the month. 1=The first week of the month. This value specifies which Sunday morning to do the "fall back". The value of this register is undefined/non-deterministic when powered up for the first time.																																																										
5:0	<p><b>FallBackHour.</b> Read-write. Reset: XXXXXXb. This Binary-Coded Decimal (BCD) value specifies which hour (24 hour mode) to do the "fall back". Fall back is usually 2 AM in United States and 1 AM in Europe. 02h=2 AM. The value of this register is undefined/non-deterministic when powered up for the first time.</p> <p><b>ValidValues:</b></p> <table> <tr> <th>Value</th><th>Description</th></tr> <tr><td>00h</td><td>2 am</td></tr> <tr><td>01h</td><td>1 am</td></tr> <tr><td>02h</td><td>2 am</td></tr> <tr><td>03h</td><td>3 am</td></tr> <tr><td>04h</td><td>4 am</td></tr> <tr><td>05h</td><td>5 am</td></tr> <tr><td>06h</td><td>6 am</td></tr> <tr><td>07h</td><td>7 am</td></tr> <tr><td>08h</td><td>8 am</td></tr> <tr><td>09h</td><td>9 am</td></tr> <tr><td>0Fh-0Ah</td><td>Reserved.</td></tr> <tr><td>10h</td><td>10 am</td></tr> <tr><td>11h</td><td>11 am</td></tr> <tr><td>12h</td><td>Noon</td></tr> <tr><td>13h</td><td>1 pm</td></tr> <tr><td>14h</td><td>2 pm</td></tr> <tr><td>15h</td><td>3 pm</td></tr> <tr><td>16h</td><td>4 pm</td></tr> <tr><td>17h</td><td>5 pm</td></tr> <tr><td>18h</td><td>6 pm</td></tr> <tr><td>19h</td><td>7 pm</td></tr> <tr><td>1Fh-1Ah</td><td>Reserved.</td></tr> <tr><td>20h</td><td>8 pm</td></tr> <tr><td>21h</td><td>9 pm</td></tr> <tr><td>22h</td><td>10 pm</td></tr> <tr><td>23h</td><td>11 pm</td></tr> <tr><td>24h</td><td>Midnight</td></tr> <tr><td>3Fh-25h</td><td>Reserved.</td></tr> </table>	Value	Description	00h	2 am	01h	1 am	02h	2 am	03h	3 am	04h	4 am	05h	5 am	06h	6 am	07h	7 am	08h	8 am	09h	9 am	0Fh-0Ah	Reserved.	10h	10 am	11h	11 am	12h	Noon	13h	1 pm	14h	2 pm	15h	3 pm	16h	4 pm	17h	5 pm	18h	6 pm	19h	7 pm	1Fh-1Ah	Reserved.	20h	8 pm	21h	9 pm	22h	10 pm	23h	11 pm	24h	Midnight	3Fh-25h	Reserved.
Value	Description																																																										
00h	2 am																																																										
01h	1 am																																																										
02h	2 am																																																										
03h	3 am																																																										
04h	4 am																																																										
05h	5 am																																																										
06h	6 am																																																										
07h	7 am																																																										
08h	8 am																																																										
09h	9 am																																																										
0Fh-0Ah	Reserved.																																																										
10h	10 am																																																										
11h	11 am																																																										
12h	Noon																																																										
13h	1 pm																																																										
14h	2 pm																																																										
15h	3 pm																																																										
16h	4 pm																																																										
17h	5 pm																																																										
18h	6 pm																																																										
19h	7 pm																																																										
1Fh-1Ah	Reserved.																																																										
20h	8 pm																																																										
21h	9 pm																																																										
22h	10 pm																																																										
23h	11 pm																																																										
24h	Midnight																																																										
3Fh-25h	Reserved.																																																										

**PMx5F\_x04 [RTCEXT FallBackMonth] (FCH::PM::RTCEXT::RTCEXTFallBackMonth)**

Read-write.

\_aliasHOSTLEGACY; PMx5F\_x04; PM=FED8\_0300h; DataPortWrite=FCH::PM::rtcextindex

Bits	Description																																
7:5	Reserved.																																
4:0	<p><b>FallBackMonth.</b> Read-write. Reset: XXXXXb. This Binary-Coded Decimal (BCD) value specifies which month to "fall back". The value of this register is undefined/non-deterministic when powered up for the first time.</p> <p><b>ValidValues:</b></p> <table> <tr> <th>Value</th><th>Description</th></tr> <tr><td>00h</td><td>October</td></tr> <tr><td>01h</td><td>January</td></tr> <tr><td>02h</td><td>February</td></tr> <tr><td>03h</td><td>March</td></tr> <tr><td>04h</td><td>April</td></tr> <tr><td>05h</td><td>May</td></tr> <tr><td>06h</td><td>June</td></tr> <tr><td>07h</td><td>July</td></tr> <tr><td>08h</td><td>August</td></tr> <tr><td>09h</td><td>September</td></tr> <tr><td>0Fh-0Ah</td><td>Reserved.</td></tr> <tr><td>10h</td><td>October</td></tr> <tr><td>11h</td><td>November</td></tr> <tr><td>12h</td><td>December</td></tr> <tr><td>1Fh-13h</td><td>Reserved.</td></tr> </table>	Value	Description	00h	October	01h	January	02h	February	03h	March	04h	April	05h	May	06h	June	07h	July	08h	August	09h	September	0Fh-0Ah	Reserved.	10h	October	11h	November	12h	December	1Fh-13h	Reserved.
Value	Description																																
00h	October																																
01h	January																																
02h	February																																
03h	March																																
04h	April																																
05h	May																																
06h	June																																
07h	July																																
08h	August																																
09h	September																																
0Fh-0Ah	Reserved.																																
10h	October																																
11h	November																																
12h	December																																
1Fh-13h	Reserved.																																

**PMx5F\_x10 [RTCEXT WeekTimerControl] (FCH::PM::RTCEXT::RTCEXTWeekTimerCtl)**

Read-write.

The 16-bit Week Timer is a battery-powered down counter timer that supports 1ms, 1 second, and 1minute resolution and auto reloads when the timer reaches 0. The WEEK\_ALRM interrupt is asserted when the timer reaches 0 and stays asserted until the timer is disabled.

\_aliasHOSTLEGACY; PMx5F\_x10; PM=FED8\_0300h; DataPortWrite=FCH::PM::rtcextindex

Bits	Description										
7:3	Reserved.										
2:1	<p><b>Resolution.</b> Read-write. Reset: XXb. This field specifies the resolution of the Week Timer counter. Before programing this bit, software should program Enable to 0 to disable the Week Timer. The value of this register is undefined/non-deterministic when powered up for the first time.</p> <p><b>ValidValues:</b></p> <table> <tr> <th>Value</th><th>Description</th></tr> <tr><td>0h</td><td>1minute</td></tr> <tr><td>1h</td><td>1second</td></tr> <tr><td>2h</td><td>1ms</td></tr> <tr><td>3h</td><td>Reserved.</td></tr> </table>	Value	Description	0h	1minute	1h	1second	2h	1ms	3h	Reserved.
Value	Description										
0h	1minute										
1h	1second										
2h	1ms										
3h	Reserved.										
0	<p><b>Enable.</b> Read-write. Reset: X. 0=Disable Week Timer. 1=Enable Week Timer. The value of this register is undefined/non-deterministic when powered up for the first time.</p>										

**PMx5F\_x11 [RTCEXT WeekTimerReloadLow] (FCH::PM::RTCEXT::RTCEXTWeekTimerReloadLo)**

Read-write.

\_aliasHOSTLEGACY; PMx5F\_x11; PM=FED8\_0300h; DataPortWrite=FCH::PM::rtcextindex

Bits	Description
7:0	<b>WeekTimerReloadLow.</b> Read-write. Reset: XXh. This field is used to program the lower 8 bits of the 16-bit WeekTimerReload register. Writing the WeekTimerReloadLow register causes the 16-bit WeekTimerReload to be written into the Week Timer. Software should program FCH::PM::RTCEXT::RTCEXTWeekTimerCtl[Enable] = 0 before writing to this register. The value of this register is undefined/non-deterministic when powered up for the first time.

**PMx5F\_x12 [RTCEXT WeekTimerReloadHigh] (FCH::PM::RTCEXT::RTCEXTWeekTimerReloadHi)**

Read-write.

\_aliasHOSTLEGACY; PMx5F\_x12; PM=FED8\_0300h; DataPortWrite=FCH::PM::rtcextindex

Bits	Description
7:0	<b>WeekTimerReloadHigh.</b> Read-write. Reset: XXh. This field is used to program the upper 8 bits of the 16-bit WeekTimerReload register. This field should be programmed before programming FCH::PM::RTCEXT::RTCEXTWeekTimerReloadLo. Software should program FCH::PM::RTCEXT::RTCEXTWeekTimerCtl[Enable] = 0 before writing to this register. The value of this register is undefined/non-deterministic when powered up for the first time.

**PMx5F\_x13 [RTCEXT WeekTimerDataLow] (FCH::PM::RTCEXT::RTCEXTWeekTimerDataLo)**

Read-only.

\_aliasHOSTLEGACY; PMx5F\_x13; PM=FED8\_0300h; DataPortWrite=FCH::PM::rtcextindex

Bits	Description
7:0	<b>WeekTimerDataLow.</b> Read-only. Reset: XXh. This field is used to read the current state of the 16-bit Week Timer. Reading from the WeekTimerDataLow register returns the lower 8 bits of the 16-bit Week Timer and causes the upper 8 bits to be latched into FCH::PM::RTCEXT::RTCEXTWeekTimerDataHi. The value of this register is undefined/non-deterministic when powered up for the first time.

**PMx5F\_x14 [RTCEXT WeekTimerDataHigh] (FCH::PM::RTCEXT::RTCEXTWeekTimerDataHi)**

Read-only.

\_aliasHOSTLEGACY; PMx5F\_x14; PM=FED8\_0300h; DataPortWrite=FCH::PM::rtcextindex

Bits	Description
7:0	<b>WeekTimerDataHigh.</b> Read-only. Reset: XXh. This field is used to read the current state of the 16-bit Week Timer. Reading from the WeekTimerDataHigh register returns the upper 8 bits of the 16-bit Week Timer latched by a previous read from FCH::PM::RTCEXT::RTCEXTWeekTimerReloadLo. The value of this register is undefined/non-deterministic when powered up for the first time.

**7.3.9.4 Power Management (PM2) Registers**

The PM2 register space is accessed through two methods:

- Indirect IO access through index/data address pair at IOCD0 [PM2\_Index] and IOCD1 [PM2\_Data]. Software first programs the offset into the index register IOCD0 [PM2\_Index] and then reads/writes to/from the data register IOCD1 [PM2\_Data].
- Memory mapped access through the AcpiMmio region. The ACPI registers range from FED8\_0000h+400h to FED8\_0000h+4FFh. See PMx04[MmioEn] for details on the AcpiMmio region.



**PM2x00000000 (FCH::PM2::fan0inputcontrol)**

Read-write. Reset: 05h.

When the fan control is not in AutoMode, the active fan duty cycle is set by LowDuty register.

When the fan is set to be controlled by the Temp\* input and set to AutoMode, the active duty cycle is controlled by the hardware automatically either in step or linear function.

(a) Step function: If step function is selected, then whenever Temp\* reaches the temperature defined by LowTemp but is less than MedTemp, the fan will be running at a duty cycle equal to LowDuty. When the temperature reaches MedTemp but is below HighTemp, the fan will be running at MedDuty. When it reaches above HighTemp, the fan will simply be running 100% duty cycle.

(b) Linear function: If linear mode is selected, the duty cycle is determined by the equations below:

When Actual Temperature &lt; LowTemp,

DutyCycle = 0

When Actual Temperature &gt; LowTemp and Actual Temperature &lt; MedTemp

DutyCycle = LowDuty

When Actual Temperature &gt; MedTemp and Actual Temperature &lt; HighTemp

DutyCycle = ((Actual Temperature - LowTemp) \* (Multiplier[5:0] + 1) &gt;&gt; Multiplier[7:6]) + LowDuty

When Actual Temperature &gt; HighTemp

DutyCycle = max or 100%

In Automode, hysteresis limit (LinearRange) is applied to keep the fan from oscillating erratically.

\_aliasHOST; PM2x00000000; PM2=FED8\_0400h

Bits	Description
7:4	Reserved.
3	<b>tworampalgorithmen</b> . Read-write. Reset: 0. <b>Description:</b> This bit is used to enable the Two Ramp Fan Control Algorithm. 0: Disable 1: Enable
2:0	<b>faninputcontrol</b> . Read-write. Reset: 5h. <b>Description:</b> 000: FanOut0 is enabled and temperature input is from internal diode. 001: FanOut0 is enabled and temperature input is from Temp0. 010: FanOut0 is enabled and temperature input is from Temp1. 011: FanOut0 is enabled and temperature input is from Temp2. 100: FanOut0 is enabled and temperature input is from Temp3. 101: FanOut0 is enabled and temperature input is from TempTsi. 110: FanOut0 is enabled and temperature input is 0. 111: FanOut0 is disabled.

**PM2x00000001 (FCH::PM2::fan0control)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000001; PM2=FED8\_0400h

Bits	Description
7:3	<b>linearadjust</b> . Read-write. Reset: 00h. Additional offset to effective duty cycle under Linear mode.
2	<b>fanpolarity</b> . Read-write. Reset: 0. <b>Description:</b> 0: FanOut0 drives low. 1: FanOut0 drives high.
1	<b>linearmode</b> . Read-write. Reset: 0. <b>Description:</b> 0: Use step function. 1: Use Linear function.
0	<b>automode</b> . Read-write. Reset: 0. Set to 1 to make FanOut0 controlled by the temperature input controlled by LowDuty0 otherwise.

**PM2x00000002 (FCH::PM2::fan0freq)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000002; PM2=FED8\_0400h

Bits	Description
7:0	<b>fanfreq.</b> Read-write. Reset: 00h. <b>Description:</b> FanOut0 frequency is programmed as follows: 00: 28.64KHz 01: 25.78KHz 02: 23.44KHz 03: 21.48KHz 04: 19.83KHz 05: 18.41KHz Any value > 05h and < F7: $\text{Freq} = 1/(\text{FreqDiv} * 2048 * 15\text{ns})$ F7: 100Hz F8: 87Hz F9: 58Hz FA: 44Hz FB: 35Hz FC: 29Hz FD: 22Hz FE: 14Hz FF: 11Hz Normally, 4-wire fan runs at 25KHz and 3-wire fan runs at 100Hz

**PM2x00000003 (FCH::PM2::lowduty0)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000003; PM2=FED8\_0400h

Bits	Description
7:0	<b>lowduty.</b> Read-write. Reset: 00h. <b>Description:</b> Fan0 Duty number when temperature is more than lowTemp0 and lower than MedTemp0. There are 256 time slots in one Fan cycle. Duty number N represents the (N+1)th time slot. Fan actively spins in time slot0~ slotN, and stops from slot-N+1 ~ slot-255. 00: Always stop FF: Full speed run

**PM2x00000004 (FCH::PM2::medduty0)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000004; PM2=FED8\_0400h

Bits	Description
7:0	<b>medduty.</b> Read-write. Reset: 00h. <b>Description:</b> Fan0 Duty number when temperature is more than MedTemp0 and lower than HighTemp0. There are 256 time slots in one Fan cycle. Duty number N represents the (N+1)th time slot. Fan actively spins in time slot0 ~ slotN, and stops from slot(N+1) ~ slot255. 00: Always stop FF: Full speed run

**PM2x00000005 (FCH::PM2::multiplier0)**

Read-write. Reset: 00h.

[NOTE] In "Two Ramp Algorithm" mode: The slope value of ramp1hi and ramp0hi is SlopeHi[7:0] = {DutySel[1:0], Multiplier[5:0]}.

SlopeHi[7:2] are the integer bits and SlopeHi[1:0] are the fractional bits of the ramp slope.

BIOS has to calculate SlopeHi[7:0] using the following equation:

$$\text{SlopeHi} = (16'hFF00 \{ \text{Med2Duty0}[7:0], 8'b0 \} ) / ( \text{HighTemp0}[15:0] \text{Hysteresis0}[15:0] \text{Med2Temp0}[15:0] )$$

For example, if our setting is:

Med2Duty0[7:0] = 25% = 40h

HighTemp0[15:0] = 90C = 5A00h

Med2Temp0[15:0] = 40C = 2800h

Hysteresis0[15:0] = 10C = 0A00h

Then:

$$\text{SlopeHi} = (FF00h \ 4000h) / (5A00h \ 0A00h \ 2800h) = 48896 / 10240 = 4.775$$

Convert the number 4.775 into our 8bit format: 00010011 (13h)

BIOS should program Multiplier0 register as 13h in this example.

\_aliasHOST; PM2x00000005; PM2=FED8\_0400h

Bits	Description
7:6	<b>dutysel.</b> Read-write. Reset: 0h. Select part of duty to be fed into fan.
5:0	<b>multiplier.</b> Read-write. Reset: 00h. Factor to calculate duty number when Fan0 is set to auto/linear mode.

**PM2x00000006 (FCH::PM2::lowtemp0lo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000006; PM2=FED8\_0400h

Bits	Description
7:0	<b>lowtemplo.</b> Read-write. Reset: 00h. LowTemp0[7:0]. Lower bits of low temperature threshold.

**PM2x00000007 (FCH::PM2::lowtemp0hi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000007; PM2=FED8\_0400h

Bits	Description
7:0	<b>lowtemphi.</b> Read-write. Reset: 00h. LowTemp0[15:8]. Higher bits of low temperature threshold.

**PM2x00000008 (FCH::PM2::medtemp0lo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000008; PM2=FED8\_0400h

Bits	Description
7:0	<b>medtemplo.</b> Read-write. Reset: 00h. MedTemp0[7:0]. Lower bits of medium temperature threshold.

**PM2x00000009 (FCH::PM2::medtemp0hi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000009; PM2=FED8\_0400h

Bits	Description
7:0	<b>medtemphi.</b> Read-write. Reset: 00h. MedTemp0[15:8]. Higher bits of medium temperature threshold.

**PM2x0000000A (FCH::PM2::hightemp0lo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000000A; PM2=FED8\_0400h

Bits	Description
7:0	<b>hightemplo.</b> Read-write. Reset: 00h. HighTemp0[7:0]. Lower bits of high temperature threshold.

**PM2x0000000B (FCH::PM2::hightemp0hi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000000B; PM2=FED8\_0400h

Bits	Description
7:0	<b>hightemphi</b> . Read-write. Reset: 00h. HighTemp0[15:8]. Higher bits of high temperature threshold.

**PM2x0000000C (FCH::PM2::linearrange0)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000000C; PM2=FED8\_0400h

Bits	Description
7:0	<b>linearrange</b> . Read-write. Reset: 00h. Variable range that Fan0 can tolerate. Fan0 will not be affected if temperature varies within this range.

**PM2x0000000D (FCH::PM2::linearholdcount0)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000000D; PM2=FED8\_0400h

Bits	Description
7:0	<b>linearholdcount</b> . Read-write. Reset: 00h. Fan cycle to be waited before duty cycle can be changed.

**PM2x0000000E (FCH::PM2::fan0hysteresis)**

Read-write. Reset: 0000h.

\_aliasHOST; PM2x0000000E; PM2=FED8\_0400h

Bits	Description
15:8	<b>hysteresishi</b> . Read-write. Reset: 00h. Configure the hysteresis value (in temperature) of the Two Ramp Fan Control Algorithm. The unit is C.
7:0	<b>hysteresislo</b> . Read-write. Reset: 00h. Reserved (This byte should be always programmed as 0)

**PM2x00000010 (FCH::PM2::fan1inputcontrol)**

Read-write. Reset: 00h.

When the fan control is not in AutoMode the active fan duty cycle is set by LowDuty register.

When the fan is set to be controlled by the Temp\* input and set to AutoMode, the active duty cycle is controlled by the hardware automatically either in step or linear function.

(a) Step function: If step function is selected, then whenever Temp\* reaches the temperature defined by LowTemp but is less than MedTemp, the fan will be running at a duty cycle equal to LowDuty. When the temperature reaches MedTemp but is below HighTemp, the fan will be running at MedDuty. When it reaches above HighTemp, the fan will simply be running 100% duty cycle.

(b) Linear function: If linear mode is selected, the duty cycle is determined by the equations below:

When Actual Temperature &lt; LowTemp,

DutyCycle = 0

When Actual Temperature &gt; LowTemp and Actual Temperature &lt; MedTemp

DutyCycle = LowDuty

When Actual Temperature &gt; MedTemp and Actual Temperature &lt; HighTemp

DutyCycle = ((Actual Temperature LowTemp) \* (Multiplier[5:0] + 1) &gt;&gt; Multiplier[7:6]) + LowDuty

When Actual Temperature &gt; HighTemp

DutyCycle = max or 100%

In Automode, hysteresis limit (LinearRange) is applied to keep the fan from oscillating erratically.

\_aliasHOST; PM2x00000010; PM2=FED8\_0400h

Bits	Description
7:3	Reserved.
2:0	<b>faninputcontrol.</b> Read-write. Reset: 0h. <b>Description:</b> 000: FanOut1 is enabled and temperature input is from Internal diode. 001: FanOut1 is enabled and temperature input is from Temp0. 010: FanOut1 is enabled and temperature input is from Temp1. 011: FanOut1 is enabled and temperature input is from Temp2. 100: FanOut1 is enabled and temperature input is from Temp3. 101: FanOut1 is enabled and temperature input is from TempTsi. 110: FanOut1 is enabled and temperature input is 0. 111: FanOut1 is disabled.

**PM2x00000011 (FCH::PM2::fan1control)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000011; PM2=FED8\_0400h

Bits	Description
7:3	<b>linearadjust.</b> Read-write. Reset: 00h. Additional offset to effective duty cycle under Linear mode.
2	<b>fanpolarity.</b> Read-write. Reset: 0. <b>Description:</b> 0: FanOut1 drives low. 1: FanOut1 drives high.
1	<b>linearmode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Use step function. 1: Use Linear function.
0	<b>automode.</b> Read-write. Reset: 0. Set to 1 to make FanOut1 controlled by the temperature input controlled by LowDuty1 otherwise.

**PM2x00000012 (FCH::PM2::fan1freq)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000012; PM2=FED8\_0400h

Bits	Description
7:0	<b>fanfreq.</b> Read-write. Reset: 00h. <b>Description:</b> FanOut1 frequency is programmed as follows: 00: 28.64KHz 01: 25.78KHz 02: 23.44KHz 03: 21.48KHz 04: 19.83KHz 05: 18.41KHz Any value > 05h and < F7 $\text{Freq} = 1/(\text{FreqDiv} * 2048 * 15\text{ns})$ F7: 100Hz F8: 87Hz F9: 58Hz FA: 44Hz FB: 35Hz FC: 29Hz FD: 22Hz FE: 14Hz FF: 11Hz Normally 4-wire fan runs at 25KHz and 3-wire fan runs at 100Hz.

**PM2x00000013 (FCH::PM2::lowduty1)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000013; PM2=FED8\_0400h

Bits	Description
7:0	<b>lowduty.</b> Read-write. Reset: 00h. <b>Description:</b> Fan1 Duty number when temperature is more than lowTemp1 and lower than MedTemp1. There are 256 time slots in one Fan cycle. Duty number N represents the (N+1)th time slot. Fan actively spins in time slot0~ slotN, and stops from slot-N+1 ~ slot-255. 00: Always stop .. FF: Full speed run

**PM2x00000014 (FCH::PM2::medduty1)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000014; PM2=FED8\_0400h

Bits	Description
7:0	<b>medduty.</b> Read-write. Reset: 00h. <b>Description:</b> Fan1 Duty number when temperature is more than MedTemp1 and lower than HighTemp1. There are 256 time slots in one Fan cycle. Duty number N represents the (N+1)th time slot. Fan actively spins in time slot0~ slotN, and stops from slot-N+1 ~ slot-255. 00: Always stop FF: Full speed run

**PM2x00000015 (FCH::PM2::multiplier1)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000015; PM2=FED8\_0400h

Bits	Description
7:6	<b>dutysel.</b> Read-write. Reset: 0h. Select part of duty to be fed into fan.
5:0	<b>multiplier.</b> Read-write. Reset: 00h. Factor to calculate duty number when FanOut1 is set to auto/linear mode.

**PM2x00000016 (FCH::PM2::lowtemp1lo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000016; PM2=FED8\_0400h

Bits	Description
7:0	<b>lowtemplo.</b> Read-write. Reset: 00h. LowTemp1[7:0]. Lower bits of low temperature threshold.

**PM2x00000017 (FCH::PM2::lowtemp1hi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000017; PM2=FED8\_0400h

Bits	Description
7:0	<b>lowtemp1hi.</b> Read-write. Reset: 00h. LowTemp1[15:8]. Higher bits of low temperature threshold.

**PM2x00000018 (FCH::PM2::medtemp1lo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000018; PM2=FED8\_0400h

Bits	Description
7:0	<b>medtemplo.</b> Read-write. Reset: 00h. MedTemp1[7:0]. Lower bits of medium temperature threshold.

**PM2x00000019 (FCH::PM2::medtemp1hi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000019; PM2=FED8\_0400h

Bits	Description
7:0	<b>medtemp1hi.</b> Read-write. Reset: 00h. MedTemp1[15:8]. Higher bits of medium temperature threshold.

**PM2x0000001A (FCH::PM2::hightemp1lo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000001A; PM2=FED8\_0400h

Bits	Description
7:0	<b>hightemplo.</b> Read-write. Reset: 00h. HighTemp1[7:0]. Lower bits of high temperature threshold.

**PM2x0000001B (FCH::PM2::hightemp1hi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000001B; PM2=FED8\_0400h

Bits	Description
7:0	<b>hightemp1hi.</b> Read-write. Reset: 00h. HighTemp1[15:8]. Higher bits of high temperature threshold.

**PM2x0000001C (FCH::PM2::linearrange1)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000001C; PM2=FED8\_0400h

Bits	Description
7:0	<b>linearrange.</b> Read-write. Reset: 00h. Variable range that FanOut1 can tolerate. FanOut1 will not be affected if temperature varies within this range.

**PM2x0000001D (FCH::PM2::linearholdcount1)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000001D; PM2=FED8\_0400h

Bits	Description
7:0	<b>linearholdcount</b> . Read-write. Reset: 00h. Fan Cycle to be waited before duty cycle can be changed.

**PM2x00000020 (FCH::PM2::fan2inputcontrol)**

Read-write. Reset: 00h.

When the fan control is not in AutoMode, the active fan duty cycle is set by LowDuty register.

When the fan is set to be controlled by the Temp\* input and set to AutoMode, the active duty cycle is controlled by the hardware automatically either in step or linear function.

(a) Step function: If step function is selected, then whenever Temp\* reaches the temperature defined by LowTemp but is less than MedTemp, the fan will be running at a duty cycle equal to LowDuty. When the temperature reaches MedTemp but is below HighTemp, the fan will be running at MedDuty. When it reaches above HighTemp, the fan will simply be running 100% duty cycle.

(b) Linear function: If linear mode is selected, the duty cycle is determined by the equations below:

When Actual Temperature &lt; LowTemp,

DutyCycle = 0

When Actual Temperature &gt; LowTemp and Actual Temperature &lt; MedTemp

DutyCycle = LowDuty

When Actual Temperature &gt; MedTemp and Actual Temperature &lt; HighTemp

DutyCycle = ((Actual Temperature LowTemp) \* (Multiplier[5:0] + 1) &gt;&gt; Multiplier[7:6]) + LowDuty

When Actual Temperature &gt; HighTemp

DutyCycle = max or 100%

In Automode, hysteresis limit (LinearRange) is applied to keep the fan from oscillating erratically.

\_aliasHOST; PM2x00000020; PM2=FED8\_0400h

Bits	Description
7:3	Reserved.
2:0	<b>faninputcontrol</b> . Read-write. Reset: 0h. <b>Description:</b> 000: FanOut2 is enabled and temperature input is from Internal diode 001: FanOut2 is enabled and temperature input is from Temp0. 010: FanOut2 is enabled and temperature input is from Temp1. 011: FanOut2 is enabled and temperature input is from Temp2. 100: FanOut2 is enabled and temperature input is from Temp3. 101: FanOut2 is enabled and temperature input is from TempTsi. 110: FanOut2 is enabled and temperature input is 0. 111: FanOut2 is disabled.



**PM2x00000021 (FCH::PM2::fan2control)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000021; PM2=FED8\_0400h

Bits	Description
7:3	<b>linearadjust.</b> Read-write. Reset: 00h. Additional offset to effective duty cycle under Linear mode.
2	<b>fanpolarity.</b> Read-write. Reset: 0. <b>Description:</b> 0: FanOut2 drives low 1: FanOut2 drives high
1	<b>linearmode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Use step function 1: Use Linear function
0	<b>automode.</b> Read-write. Reset: 0. Set to 1 to make FanOut2 controlled by the temperature input controlled by LowDuty2 otherwise.

**PM2x00000022 (FCH::PM2::fan2freq)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000022; PM2=FED8\_0400h

Bits	Description
7:0	<b>fanfreq.</b> Read-write. Reset: 00h. <b>Description:</b> FanOut2 frequency is programmed as follows: 00: 28.64KHz 01: 25.78KHz 02: 23.44KHz 03: 21.48KHz 04: 19.83KHz 05: 18.41KHz Any value > 05h and < F7: $\text{Freq} = 1/(\text{FreqDiv} * 2048 * 15\text{ns})$ F7: 100Hz F8: 87Hz F9: 58Hz FA: 44Hz FB: 35Hz FC: 29Hz FD: 22Hz FE: 14Hz FF: 11Hz Normally 4-wire fan runs at 25KHz and 3-wire fan runs at 100Hz

**PM2x00000023 (FCH::PM2::lowduty2)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000023; PM2=FED8\_0400h

Bits	Description
7:0	<b>lowduty.</b> Read-write. Reset: 00h. <b>Description:</b> Fan2 Duty number when temperature is more than lowTemp2 and lower than MedTemp2. There are 256 time slots in one fan cycle. Duty number N represents (N+1)th time slot. Fan actively spins in time slot0~ slotN, and stops from slot-N+1 ~ slot-255. 00: Always stop FF: Full speed run

**PM2x00000024 (FCH::PM2::medduty2)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000024; PM2=FED8\_0400h

Bits	Description
7:0	<b>medduty</b> . Read-write. Reset: 00h. <b>Description:</b> FanOut2 Duty number when temperature is more than MedTemp2 and lower than HighTemp2. There are 256 time slots in one Fan cycle. Duty number N represents (N+1)th time slot. Fan actively spins in time slot0~ slotN, and stops from slot-N+1 ~ slot-255. 00: Always stop FF: Full speed run

**PM2x00000025 (FCH::PM2::multiplier2)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000025; PM2=FED8\_0400h

Bits	Description
7:6	<b>dutysel</b> . Read-write. Reset: 0h. Select part of duty to be fed into fan.
5:0	<b>multiplier</b> . Read-write. Reset: 00h. Factor to calculate duty number when FanOut2 is set to auto/linear mode.

**PM2x00000026 (FCH::PM2::lowtemp2lo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000026; PM2=FED8\_0400h

Bits	Description
7:0	<b>lowtemplo</b> . Read-write. Reset: 00h. LowTemp2[7:0]. Lower bits of low temperature threshold.

**PM2x00000027 (FCH::PM2::lowtemp2hi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000027; PM2=FED8\_0400h

Bits	Description
7:0	<b>lowtemp2hi</b> . Read-write. Reset: 00h. LowTemp2[15:8]. Higher bits of low temperature threshold.

**PM2x00000028 (FCH::PM2::medtemp2lo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000028; PM2=FED8\_0400h

Bits	Description
7:0	<b>medtemplo</b> . Read-write. Reset: 00h. MedTemp2[7:0]. Lower bits of medium temperature threshold.

**PM2x00000029 (FCH::PM2::medtemp2hi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000029; PM2=FED8\_0400h

Bits	Description
7:0	<b>medtemp2hi</b> . Read-write. Reset: 00h. MedTemp2[15:8]. Higher bits of medium temperature threshold.

**PM2x0000002A (FCH::PM2::hightemp2lo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000002A; PM2=FED8\_0400h

Bits	Description
7:0	<b>hightemplo</b> . Read-write. Reset: 00h. HighTemp2[7:0]. Lower bits of high temperature threshold.

**PM2x0000002B (FCH::PM2::hightemp2hi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000002B; PM2=FED8\_0400h

Bits	Description
7:0	<b>hightemp2hi</b> . Read-write. Reset: 00h. HighTemp2[15:8]. Higher bits of high temperature threshold.

**PM2x0000002C (FCH::PM2::linearrange2)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000002C; PM2=FED8\_0400h

Bits	Description
7:0	<b>linearrange</b> . Read-write. Reset: 00h. Variable range that FanOut2 can tolerate. FanOut2 will not be affected if temperature varies within this range.

**PM2x0000002D (FCH::PM2::linearholdcount2)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000002D; PM2=FED8\_0400h

Bits	Description
7:0	<b>linearholdcount</b> . Read-write. Reset: 00h. Fan Cycle to be waited before duty cycle can be changed.

**PM2x00000030 (FCH::PM2::fan3inputcontrol)**

Read-write. Reset: 00h.

When the fan control is not in AutoMode, the active fan duty cycle is set by LowDuty register.

When the fan is set to be controlled by the Temp\* input and set to AutoMode, the active duty cycle is controlled by the hardware automatically either in step or linear function.

(a) Step function: If step function is selected, then whenever Temp\* reaches the temperature defined by LowTemp but is less than MedTemp, the fan will be running at a duty cycle equal to LowDuty. When the temperature reaches MedTemp but is below HighTemp, the fan will be running at MedDuty. When it reaches above HighTemp, the fan will simply be running 100% duty cycle.

(b) Linear function: If linear mode is selected, the duty cycle is determined by the equations below:

When Actual Temperature &lt; LowTemp,

DutyCycle = 0

When Actual Temperature &gt; LowTemp and Actual Temperature &lt; MedTemp

DutyCycle = LowDuty

When Actual Temperature &gt; MedTemp and Actual Temperature &lt; HighTemp

DutyCycle = ((Actual Temperature LowTemp) \* (Multiplier[5:0] + 1) &gt;&gt; Multiplier[7:6]) + LowDuty

When Actual Temperature &gt; HighTemp

DutyCycle = max or 100%

In Automode, hysteresis limit (LinearRange) is applied to keep the fan from oscillating erratically.

\_aliasHOST; PM2x00000030; PM2=FED8\_0400h

Bits	Description
7:3	Reserved.
2:0	<b>faninputcontrol</b> . Read-write. Reset: 0h. <b>Description:</b> 000: FanOut3 is enabled and temperature input is from Internal diode. 001: FanOut3 is enabled and temperature input is from Temp0. 010: FanOut3 is enabled and temperature input is from Temp1. 011: FanOut3 is enabled and temperature input is from Temp2. 100: FanOut3 is enabled and temperature input is from Temp3. 101: FanOut3 is enabled and temperature input is from TempTsi. 110: FanOut3 is enabled and temperature input is 0. 111: FanOut3 is disabled.

**PM2x00000031 (FCH::PM2::fan3control)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000031; PM2=FED8\_0400h

Bits	Description
7:3	<b>linearadjust.</b> Read-write. Reset: 00h. Additional offset to effective duty cycle under Linear mode.
2	<b>fanpolarity.</b> Read-write. Reset: 0. <b>Description:</b> 0: FanOut3 drives low 1: FanOut3 drives high
1	<b>linearmode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Use step function 1: Use linear function
0	<b>automode.</b> Read-write. Reset: 0. Set to 1 to make FanOut3 controlled by the temperature input controlled by LowDuty3 otherwise.

**PM2x00000032 (FCH::PM2::fan3freq)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000032; PM2=FED8\_0400h

Bits	Description
7:0	<b>fanfreq.</b> Read-write. Reset: 00h. <b>Description:</b> FanOut3 frequency is programmed as follows: 00: 28.64KHz 01: 25.78KHz 02: 23.44KHz 03: 21.48KHz 04: 19.83KHz 05: 18.41KHz Any value > 05h and < F7: $\text{Freq} = 1/(\text{FreqDiv} * 2048 * 15\text{ns})$ F7: 100Hz F8: 87Hz F9: 58Hz FA: 44Hz FB: 35Hz FC: 29Hz FD: 22Hz FE: 14Hz FF: 11Hz Normally 4-wire fan runs at 25KHz and 3-wire fan runs at 100Hz

**PM2x00000033 (FCH::PM2::lowduty3)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000033; PM2=FED8\_0400h

Bits	Description
7:0	<b>lowduty.</b> Read-write. Reset: 00h. <b>Description:</b> FanOut3 Duty number when temperature is more than lowTemp3 and lower than MedTemp3. There are 256 time slots in one Fan cycle. Duty number N represents (N+1)th time slot. Fan actively spins in time slot0~ slotN, and stops from slot-N+1 ~ slot-255. 00: Always stop ... FF: Full speed run

**PM2x00000034 (FCH::PM2::medduty3)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000034; PM2=FED8\_0400h

Bits	Description
7:0	<b>medduty</b> . Read-write. Reset: 00h. <b>Description:</b> FanOut3 Duty number when temperature is more than MedTemp3 and lower than HighTemp3. There are 256 time slots in one Fan cycle. Duty number N represents (N+1)th time slot. Fan actively spins in time slot0~ slotN, and stops from slot-N+1 ~ slot-255. 00: Always stop FF: Full speed run

**PM2x00000035 (FCH::PM2::multiplier3)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000035; PM2=FED8\_0400h

Bits	Description
7:6	<b>dutysel</b> . Read-write. Reset: 0h. Select part of duty to be fed into fan.
5:0	<b>multiplier</b> . Read-write. Reset: 00h. Factor to calculate duty number when FanOut3 is set to auto/linear mode.

**PM2x00000036 (FCH::PM2::lowtemp3lo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000036; PM2=FED8\_0400h

Bits	Description
7:0	<b>lowtemplo</b> . Read-write. Reset: 00h. LowTemp3[7:0]. Lower bits of low temperature threshold.

**PM2x00000037 (FCH::PM2::lowtemp3hi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000037; PM2=FED8\_0400h

Bits	Description
7:0	<b>lowtemphi</b> . Read-write. Reset: 00h. LowTemp3[15:8]. Higher bits of low temperature threshold.

**PM2x00000038 (FCH::PM2::medtemp3lo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000038; PM2=FED8\_0400h

Bits	Description
7:0	<b>medtemplo</b> . Read-write. Reset: 00h. MedTemp3[7:0]. Lower bits of medium temperature threshold.

**PM2x00000039 (FCH::PM2::medtemp3hi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000039; PM2=FED8\_0400h

Bits	Description
7:0	<b>medtemphi</b> . Read-write. Reset: 00h. MedTemp3[15:8]. Higher bits of medium temperature threshold.

**PM2x0000003A (FCH::PM2::hightemp3lo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000003A; PM2=FED8\_0400h

Bits	Description
7:0	<b>hightemplo</b> . Read-write. Reset: 00h. HighTemp3[7:0]. Lower bits of high temperature threshold.

**PM2x0000003B (FCH::PM2::hightemp3hi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000003B; PM2=FED8\_0400h

Bits	Description
7:0	<b>hightemphi</b> . Read-write. Reset: 00h. HighTemp3[15:8]. Higher bits of high temperature threshold.

**PM2x0000003C (FCH::PM2::linearrange3)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000003C; PM2=FED8\_0400h

Bits	Description
7:0	<b>linearrange</b> . Read-write. Reset: 00h. Variable range that FanOut3 can tolerate. FanOut3 will not be affected if temperature varies within this range.

**PM2x0000003D (FCH::PM2::linearholdcount3)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000003D; PM2=FED8\_0400h

Bits	Description
7:0	<b>linearholdcount</b> . Read-write. Reset: 00h. Fan Cycle to be waited before duty cycle can be changed.

**PM2x00000040 (FCH::PM2::fan4inputcontrol)**

Read-write. Reset: 00h.

When the fan control is not in AutoMode, the active fan duty cycle is set by LowDuty register.

When the fan is set to be controlled by the Temp\* input and set to AutoMode, the active duty cycle is controlled by the hardware automatically either in step or linear function.

(a) Step function: If step function is selected, then whenever Temp\* reaches the temperature defined by LowTemp but is less than MedTemp, the fan will be running at a duty cycle equal to LowDuty. When the temperature reaches MedTemp but is below HighTemp, the fan will be running at MedDuty. When it reaches above HighTemp, the fan will simply be running 100% duty cycle.

(b) Linear function: If linear mode is selected, the duty cycle is determined by the equations below:

When Actual Temperature &lt; LowTemp,

DutyCycle = 0

When Actual Temperature &gt; LowTemp and Actual Temperature &lt; MedTemp

DutyCycle = LowDuty

When Actual Temperature &gt; MedTemp and Actual Temperature &lt; HighTemp

DutyCycle = ((Actual Temperature - LowTemp) \* (Multiplier[5:0] + 1) &gt;&gt; Multiplier[7:6]) + LowDuty

When Actual Temperature &gt; HighTemp

DutyCycle = max or 100%

In Automode, hysteresis limit (LinearRange) is applied to keep the fan from oscillating erratically.

\_aliasHOST; PM2x00000040; PM2=FED8\_0400h

Bits	Description
7:3	Reserved.
2:0	<b>faninputcontrol</b> . Read-write. Reset: 0h. <b>Description:</b> 000: FanOut4 is enabled and temperature input is from Internal diode. 001: FanOut4 is enabled and temperature input is from Temp0. 010: FanOut4 is enabled and temperature input is from Temp1. 011: FanOut4 is enabled and temperature input is from Temp2. 100: FanOut4 is enabled and temperature input is from Temp3. 101: FanOut3 is enabled and temperature input is from TempTsi. 110: FanOut4 is enabled and temperature input is 0. 111: FanOut4 is disabled.

**PM2x00000041 (FCH::PM2::fan4control)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000041; PM2=FED8\_0400h

Bits	Description
7:3	<b>linearadjust.</b> Read-write. Reset: 00h. Additional offset to effective duty cycle under linear mode.
2	<b>fanpolarity.</b> Read-write. Reset: 0. <b>Description:</b> 0: FanOut4 drives low 1: FanOut4 drives high
1	<b>linearmode.</b> Read-write. Reset: 0. <b>Description:</b> 0: Use step function 1: Use linear function
0	<b>automode.</b> Read-write. Reset: 0. Set to 1 to make FanOut4 controlled by the temperature input controlled by LowDuty4 otherwise.

**PM2x00000042 (FCH::PM2::fan4freq)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000042; PM2=FED8\_0400h

Bits	Description
7:0	<b>fanfreq.</b> Read-write. Reset: 00h. <b>Description:</b> FanOut4 frequency is programmed as follows: 00: 28.64KHz 01: 25.78KHz 02: 23.44KHz 03: 21.48KHz 04: 19.83KHz 05: 18.41KHz Any value > 05h and < F7 Freq = 1/(FreqDiv * 2048 * 15ns) F7: 100Hz F8: 87Hz F9: 58Hz FA: 44Hz FB: 35Hz FC: 29Hz FD: 22Hz FE: 14Hz FF: 11Hz Normally 4-wire fan runs at 25KHz and 3-wire fan runs at 100Hz.

**PM2x00000043 (FCH::PM2::lowduty4)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000043; PM2=FED8\_0400h

Bits	Description
7:0	<b>lowduty.</b> Read-write. Reset: 00h. <b>Description:</b> FanOut4 Duty number when temperature is more than LowTemp4 and lower than MedTemp4. There are 256 time slots in one Fan cycle. Duty number N represents (N+1)th time slot. Fan actively spins in time slot0~ slotN, and stops from slot-N+1 ~ slot-255. 00: Always stop. FF: Full speed run.

**PM2x00000044 (FCH::PM2::medduty4)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000044; PM2=FED8\_0400h

Bits	Description
7:0	<b>medduty</b> . Read-write. Reset: 00h. <b>Description:</b> FanOut4 Duty number when temperature is more than MedTemp4 and lower than HighTemp4. There are 256 time slots in one Fan cycle. Duty number N represents (N+1)th time slot. Fan actively spins in time slot0~ slotN, and stops from slot-N+1 ~ slot-255. 00: Always stop FF: Full speed run

**PM2x00000045 (FCH::PM2::multiplier4)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000045; PM2=FED8\_0400h

Bits	Description
7:6	<b>dutysel</b> . Read-write. Reset: 0h. Select part of duty to be fed into fan.
5:0	<b>multiplier</b> . Read-write. Reset: 00h. Factor to calculate duty number when FanOut4 is set to auto/linear mode.

**PM2x00000046 (FCH::PM2::lowtemp4lo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000046; PM2=FED8\_0400h

Bits	Description
7:0	<b>lowtemplo</b> . Read-write. Reset: 00h. LowTemp4[7:0]. Lower bits of low temperature threshold.

**PM2x00000047 (FCH::PM2::lowtemp4hi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000047; PM2=FED8\_0400h

Bits	Description
7:0	<b>lowtemphi</b> . Read-write. Reset: 00h. LowTemp4[15:8]. Higher bits of low temperature threshold.

**PM2x00000048 (FCH::PM2::medtemp4lo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000048; PM2=FED8\_0400h

Bits	Description
7:0	<b>medtemplo</b> . Read-write. Reset: 00h. MedTemp4[7:0]. Lower bits of medium temperature threshold.

**PM2x00000049 (FCH::PM2::medtemp4hi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000049; PM2=FED8\_0400h

Bits	Description
7:0	<b>medtemphi</b> . Read-write. Reset: 00h. MedTemp4[15:8]. Higher bits of medium temperature threshold.

**PM2x0000004A (FCH::PM2::hightemp4lo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000004A; PM2=FED8\_0400h

Bits	Description
7:0	<b>hightemplo</b> . Read-write. Reset: 00h. HighTemp4[7:0]. Lower bits of high temperature threshold.

**PM2x0000004B (FCH::PM2::hightemp4hi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000004B; PM2=FED8\_0400h

Bits	Description
7:0	<b>hightemphi</b> . Read-write. Reset: 00h. HighTemp4[15:8]. Higher bits of high temperature threshold.



**PM2x0000004C (FCH::PM2::linearrange4)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000004C; PM2=FED8\_0400h

Bits	Description
7:0	<b>linearrange</b> . Read-write. Reset: 00h. Variable range that FanOut4 can tolerate. FanOut4 will not be affected if temperature varies within this range.

**PM2x0000004D (FCH::PM2::linearholdcount4)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000004D; PM2=FED8\_0400h

Bits	Description
7:0	<b>linearholdcount</b> . Read-write. Reset: 00h. Fan Cycle to be waited before duty cycle can be changed.

**PM2x00000050 (FCH::PM2::med2temp0lo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000050; PM2=FED8\_0400h

Bits	Description
7:0	<b>med2temp0_7_0</b> . Read-write. Reset: 00h. <b>Description:</b> This register is used only when TwoRampAlgorithmEn is 1. It specifies the lower byte of the temperature value of the turning point on the ramp.

**PM2x00000051 (FCH::PM2::med2temp0hi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000051; PM2=FED8\_0400h

Bits	Description
7:0	<b>med2temp0_15_8</b> . Read-write. Reset: 00h. <b>Description:</b> This register is used only when TwoRampAlgorithmEn is 1. It specifies the higher byte of the temperature value of the turning point on the ramp.

**PM2x00000052 (FCH::PM2::med2duty0)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000052; PM2=FED8\_0400h

Bits	Description
7:0	<b>med2duty0_7_0</b> . Read-write. Reset: 00h. <b>Description:</b> This register is used only when TwoRampAlgorithmEn is 1. It specifies the fan duty value of the turning point on the ramp.

**PM2x00000053 (FCH::PM2::multiplier2\_0)**

Read-write. Reset: 00h.

[NOTE] In "Two Ramp Algorithm" mode: This register specify the slope value of ramp1lo and ramp0lo.

Multiplier2\_0[7:2] are the integer bits and Multiplier2\_0[1:0] are the fractional bits of the ramp slope.

BIOS has to calculate Multiplier2\_0[7:0] using the following equation:

$$\text{Multiplier2\_0} = (\{\text{Med2Duty0}[7:0], 8'b0\} \{\text{LowDuty0}[7:0], 8'b0\}) / (\text{Med2Temp0}[15:0] \text{MedTemp0}[15:0])$$

For example, if our setting is:

Med2Duty0[7:0] = 50% = 80h

LowDuty0[7:0] = 25% = 40h

Med2Temp0[15:0] = 70C = 4600h

MedTemp0[15:0] = 40C = 2800h

Then:

$$\text{Multiplier2\_0} = (8000h \ 4000h) / (4600h \ 2800h) = 16384 / 7680 = 2.133$$

Convert the number 4.775 into our 8bit format: 00001000b (08h)

BIOS should program Multiplier0 register as 08h in this example.

\_aliasHOST; PM2x00000053; PM2=FED8\_0400h

Bits	Description
7:0	<b>multiplier2_0_7_0.</b> Read-write. Reset: 00h. <b>Description:</b> This register is used only when TwoRampAlgorithmEn is 1. It specifies the slope value of ramp1lo and ramp0lo. Bit 7~2 are integer bits. Bit 1~0 are fractional bits.

**PM2x00000060 (FCH::PM2::fanstatus)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000060; PM2=FED8\_0400h

Bits	Description
7:5	Reserved.
4	<b>fan4speedtooslow.</b> Read-write. Reset: 0. Indicates whether Fan4 runs slower than the value in the Fan4SpeedLimit. Write to 1 to clear.
3	<b>fan3speedtooslow.</b> Read-write. Reset: 0. Indicates whether Fan3 runs slower than the value in the Fan3SpeedLimit. Write to 1 to clear.
2	<b>fan2speedtooslow.</b> Read-write. Reset: 0. Indicates whether Fan2 runs slower than the value in the Fan2SpeedLimit. Write to 1 to clear.
1	<b>fan1speedtooslow.</b> Read-write. Reset: 0. Indicates whether Fan1 runs slower than the value in the Fan1SpeedLimit. Write to 1 to clear.
0	<b>fan0speedtooslow.</b> Read-write. Reset: 0. Indicates whether Fan0 runs slower than the value in the Fan0SpeedLimit. Write to 1 to clear.

**PM2x00000061 (FCH::PM2::fanintroutelo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000061; PM2=FED8\_0400h

Bits	Description
7:6	<b>fan3introute.</b> Read-write. Reset: 0h. <b>Description:</b> 01: SMI 10: SMI or SCI according GEVENT13 routing Others: No SCI/SMI generated
5:4	<b>fan2introute.</b> Read-write. Reset: 0h. <b>Description:</b> 01: SMI 10: SMI or SCI according GEVENT13 routing Others: No SCI/SMI generated
3:2	<b>fan1introute.</b> Read-write. Reset: 0h. <b>Description:</b> 01: SMI 10: SMI or SCI according GEVENT13 routing Others: No SCI/SMI generated
1:0	<b>fan0introute.</b> Read-write. Reset: 0h. <b>Description:</b> 01: SMI 10: SMI or SCI according GEVENT13 routing Others: No SCI/SMI generated

**PM2x00000062 (FCH::PM2::fanintroutehi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000062; PM2=FED8\_0400h

Bits	Description
7:2	Reserved.
1:0	<b>fan4introute.</b> Read-write. Reset: 0h. <b>Description:</b> 01: SMI 10: SMI or SCI according GEVENT13 routing Others: No SCI/SMI generated

**PM2x00000063 (FCH::PM2::samplefreqdiv)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000063; PM2=FED8\_0400h

Bits	Description
7:4	<b>linearrangeoutlimit_4_1.</b> Read-write. Reset: 0h. <b>Description:</b> LinearRangeOutLimit[7:0] = {000b, LinearRangeOutLimit[4:1], 1b} This parameter defines how close the fan duty will follow the "target duty cycle". It is NOT the same with LinearRange, which works like a hysteresis and used when fan duty is not changing. LinearRangeOutLimit is only used when the fan duty is changing.
3	Reserved.
2	<b>fanlinearenhanceen2.</b> Read-write. Reset: 0. When this bit is set, the positive hysteresis of fan duty is removed. In other words, LinearRange only apply to the negative direction when this bit is 1. As a result, the fan duty will increase once the temperature is increased instead of waiting for a hysteresis.
1:0	<b>samplefreqdiv.</b> Read-write. Reset: 0h. <b>Description:</b> These bits determine the sampling rate of Fan Speed. 00: Base(22.5KHz) 01: Base(22.5KHz)/2 10: Base(22.5KHz)/4 11: Base(22.5KHz)/8

**PM2x00000064 (FCH::PM2::fandebouncecounterlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000064; PM2=FED8\_0400h

Bits	Description
7:0	<b>fandebouncecounterlo</b> . Read-write. Reset: 00h. Specify low 8 bits of the debounced counter when measuring Fan Speed

**PM2x00000065 (FCH::PM2::fandebouncecounterhi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000065; PM2=FED8\_0400h

Bits	Description
7:0	<b>fandebouncecounterhi</b> . Read-write. Reset: 00h. Specify high 8 bits of the debounced counter when measuring Fan Speed

**PM2x00000066 (FCH::PM2::fan0detectorcontrol)**

Read-write.

\_aliasHOST; PM2x00000066; PM2=FED8\_0400h

Bits	Description
7:5	Reserved.
4	<b>shutdownenable</b> . Read-write. Reset: 0. If set to 1, the machine can be shutdown if the Fan0 Status remains for more than 4 seconds.
3:2	Reserved.
1	<b>useaverage</b> . Read-write. Reset: 0. <b>Description:</b> 0: Not to average Fan0 speed 1: Average Fan0 speed
0	<b>fan0detectorenable</b> . Read-write. Reset: 0. <b>Description:</b> 0: Disable Fan0 speed measurement 1: Enable Fan0 speed measurement

**PM2x00000067 (FCH::PM2::fan0speedlimitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000067; PM2=FED8\_0400h

Bits	Description
7:0	<b>fan0speedlimitlo</b> . Read-write. Reset: 00h. Lower 8 bits of Fan0SpeedLimit to set threshold when Fan0 speed is below it.

**PM2x00000068 (FCH::PM2::fan0speedlimithi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000068; PM2=FED8\_0400h

Bits	Description
7:0	<b>fan0speedlimithi</b> . Read-write. Reset: 00h. Higher 8 bits of Fan0SpeedLimit to set threshold when Fan0 speed is below it.

**PM2x00000069 (FCH::PM2::fan0speedlo)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x00000069; PM2=FED8\_0400h

Bits	Description
7:1	<b>fan0speedlo</b> . Read-only. Reset: 00h. Fan0Speed [7:0]
0	Reserved.

**PM2x0000006A (FCH::PM2::fan0speedhi)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x0000006A; PM2=FED8\_0400h

Bits	Description
7:0	<b>fanspeed.</b> Read-only. Reset: 00h. Fan0Speed[15:8]

**PM2x0000006B (FCH::PM2::fan1detectorcontrol)**

Read-write.

\_aliasHOST; PM2x0000006B; PM2=FED8\_0400h

Bits	Description
7:5	Reserved.
4	<b>shutdownenable.</b> Read-write. Reset: 0. If set to 1, the machine can be shutdown if the Fan1 Status remains for more than 4 seconds.
3:2	Reserved.
1	<b>useaverage.</b> Read-write. Reset: 0. <b>Description:</b> 0: Not to average Fan1 speed 1: Average Fan1 speed
0	<b>fanetectorenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable Fan1 speed measurement 1: Enable Fan1 speed measurement

**PM2x0000006C (FCH::PM2::fan1speedlimitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000006C; PM2=FED8\_0400h

Bits	Description
7:0	<b>fanspeedlimit.</b> Read-write. Reset: 00h. Lower 8 bits of Fan1SpeedLimit to set threshold when Fan1 speed is below it.

**PM2x0000006D (FCH::PM2::fan1speedlimithi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000006D; PM2=FED8\_0400h

Bits	Description
7:0	<b>fanspeedlimit.</b> Read-write. Reset: 00h. Higher 8 bits of Fan1SpeedLimit to set threshold when Fan1 speed is below it.

**PM2x0000006E (FCH::PM2::fan1speedlo)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x0000006E; PM2=FED8\_0400h

Bits	Description
7:1	<b>fanspeed.</b> Read-only. Reset: 00h. Fan1Speed[7:0]
0	Reserved.

**PM2x0000006F (FCH::PM2::fan1speedhi)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x0000006F; PM2=FED8\_0400h

Bits	Description
7:0	<b>fanspeed.</b> Read-only. Reset: 00h. Fan1Speed[15:8]

**PM2x00000070 (FCH::PM2::fan2detectorcontrol)**

Read-write.

\_aliasHOST; PM2x00000070; PM2=FED8\_0400h

Bits	Description
7:5	Reserved.
4	<b>shutdownenable</b> . Read-write. Reset: 0. If set to, the machine can be shutdown if the Fan2 Status remains for more than 4 seconds.
3:2	Reserved.
1	<b>useaverage</b> . Read-write. Reset: 0. <b>Description:</b> 0: Not to average Fan2 speed 1: Average Fan2 speed
0	<b>fanetectorenable</b> . Read-write. Reset: 0. <b>Description:</b> 0: Disable Fan2 speed measurement 1: Enable Fan2 speed measurement

**PM2x00000071 (FCH::PM2::fan2speedlimitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000071; PM2=FED8\_0400h

Bits	Description
7:0	<b>fanspeedlimit</b> . Read-write. Reset: 00h. Lower 8 bits of Fan2SpeedLimit to set threshold when Fan2 speed is below it.

**PM2x00000072 (FCH::PM2::fan2speedlimithi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000072; PM2=FED8\_0400h

Bits	Description
7:0	<b>fanspeedlimit</b> . Read-write. Reset: 00h. Higher 8 bits of Fan2SpeedLimit to set threshold when Fan2 speed is below it.

**PM2x00000073 (FCH::PM2::fan2speedlo)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x00000073; PM2=FED8\_0400h

Bits	Description
7:1	<b>fanspeed</b> . Read-only. Reset: 00h. Fan2Speed[7:0]
0	Reserved.

**PM2x00000074 (FCH::PM2::fan2speedhi)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x00000074; PM2=FED8\_0400h

Bits	Description
7:0	<b>fanspeed</b> . Read-only. Reset: 00h. Fan2Speed[15:8]

**PM2x00000075 (FCH::PM2::fan3detectorcontrol)**

Read-write.

\_aliasHOST; PM2x00000075; PM2=FED8\_0400h

Bits	Description
7:5	Reserved.
4	<b>shutdownenable.</b> Read-write. Reset: 0. If set to, the machine can be shutdown if the Fan3 Status remains for more than 4 seconds.
3:2	Reserved.
1	<b>useaverage.</b> Read-write. Reset: 0. <b>Description:</b> 0: Not to average Fan3 speed 1: Average Fan3 speed
0	<b>fanetectorenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable Fan3 speed measurement 1: Enable Fan3 speed measurement

**PM2x00000076 (FCH::PM2::fan3speedlimitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000076; PM2=FED8\_0400h

Bits	Description
7:0	<b>fanspeedlimit.</b> Read-write. Reset: 00h. Lower 8 bits of Fan3SpeedLimit to set threshold when Fan3 speed is below it.

**PM2x00000077 (FCH::PM2::fan3speedlimithi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000077; PM2=FED8\_0400h

Bits	Description
7:0	<b>fanspeedlimit.</b> Read-write. Reset: 00h. Higher 8 bits of Fan3SpeedLimit to set threshold when Fan3 speed is below it.

**PM2x00000078 (FCH::PM2::fan3speedlo)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x00000078; PM2=FED8\_0400h

Bits	Description
7:1	<b>fanspeed.</b> Read-only. Reset: 00h. Fan3Speed[7:0]
0	Reserved.

**PM2x00000079 (FCH::PM2::fan3speedhi)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x00000079; PM2=FED8\_0400h

Bits	Description
7:0	<b>fanspeed.</b> Read-only. Reset: 00h. Fan3Speed[15:8]

**PM2x0000007A (FCH::PM2::fan4detectorcontrol)**

Read-write.

\_aliasHOST; PM2x0000007A; PM2=FED8\_0400h

Bits	Description
7:5	Reserved.
4	<b>shutdownenable</b> . Read-write. Reset: 0. If set to, the machine can be shutdown if the Fan4 Status remains for more than 4 seconds.
3:2	Reserved.
1	<b>useaverage</b> . Read-write. Reset: 0. <b>Description:</b> 0: Not to average Fan4 speed 1: Average Fan4 speed
0	<b>fanetectorenable</b> . Read-write. Reset: 0. <b>Description:</b> 0: Disable Fan4 speed measurement 1: Enable Fan4 speed measurement

**PM2x0000007B (FCH::PM2::fan4speedlimitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000007B; PM2=FED8\_0400h

Bits	Description
7:0	<b>fanspeedlimit</b> . Read-write. Reset: 00h. Lower 8 bits of Fan4SpeedLimit to set threshold when Fan4 speed is below it.

**PM2x0000007C (FCH::PM2::fan4speedlimithi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000007C; PM2=FED8\_0400h

Bits	Description
7:0	<b>fanspeedlimit</b> . Read-write. Reset: 00h. Higher 8 bits of Fan4SpeedLimit to set threshold when Fan4 speed is below it.

**PM2x0000007D (FCH::PM2::fan4speedlo)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x0000007D; PM2=FED8\_0400h

Bits	Description
7:1	<b>fanspeed</b> . Read-only. Reset: 00h. Fan4Speed[7:0]
0	Reserved.

**PM2x0000007E (FCH::PM2::fan4speedhi)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x0000007E; PM2=FED8\_0400h

Bits	Description
7:0	<b>fanspeed</b> . Read-only. Reset: 00h. Fan4Speed[15:8]

**PM2x0000008A (FCH::PM2::temptсило)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000008A; PM2=FED8\_0400h

Bits	Description
7:0	<b>temptсило</b> . Read-write. Reset: 00h. TempTsi[7:0].



**PM2x0000008B (FCH::PM2::temptsihi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000008B; PM2=FED8\_0400h

Bits	Description
7:0	<b>temptsihi</b> . Read-write. Reset: 00h. TempTsi[15:8].

**PM2x0000008C (FCH::PM2::temptsilimitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000008C; PM2=FED8\_0400h

Bits	Description
7:0	<b>temptsilimitlo</b> . Read-write. Reset: 00h. TempTsiLimit[7:0]

**PM2x0000008D (FCH::PM2::temptsilimithi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000008D; PM2=FED8\_0400h

Bits	Description
7:0	<b>temptsilimithi</b> . Read-write. Reset: 00h. TempTsiLimit[15:8]

**PM2x0000008E (FCH::PM2::temptkichangelimit)**

Read-write. Reset: 00h.

When Temp(new) &gt; Temp(old) + [TempChangeLimit &lt;&lt; 6]

Temp = Temp(old) + [TempChangeLimit &lt;&lt; 6]

When Temp(new) &lt; Temp(old) [TempChangeLimit &lt;&lt; 6]

Temp = Temp(old) [TempChangeLimit &lt;&lt; 6]

When Temp(new) &lt;= Temp(old) + [TempChangeLimit &lt;&lt; 6] and &gt;= Temp(old) [TempChangeLimit &lt;&lt; 6]

Temp = Temp(new)

\_aliasHOST; PM2x0000008E; PM2=FED8\_0400h

Bits	Description
7:0	<b>temptkichangelimit</b> . Read-write. Reset: 00h. Filtering is applied to {TempTsiHi, TempTsiLo} if TempChangeLimit is nonzero.

**PM2x0000008F (FCH::PM2::tempwe)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000008F; PM2=FED8\_0400h

Bits	Description
7:6	Reserved.
5	<b>temptsiwe.</b> Read-write. Reset: 0. <b>Description:</b> 0: TempTsi is a read-only register, which is the result from TempTsi Sensor. 1: Only Host or EC can write TempTsi register. The TempTsi Sensor result doesn't affect TempTsi.
4	<b>temp3we.</b> Read-write. Reset: 0. <b>Description:</b> 0: Temp3 is a read-only register, which is the result from Temp3 Sensor. 1: Only Host or EC can write Temp3 register. The Temp3 Sensor result doesn't affect Temp3.
3	<b>temp2we.</b> Read-write. Reset: 0. <b>Description:</b> 0: Temp2 is a read-only register, which is the result from Temp2 Sensor. 1: Only Host or EC can write Temp2 register. The Temp2 Sensor result doesn't affect Temp2.
2	<b>temp1we.</b> Read-write. Reset: 0. <b>Description:</b> 0: Temp1 is a read-only register, which is the result from Temp1 Sensor. 1: Only Host or EC can write Temp1 register. The Temp1 Sensor result doesn't affect Temp1.
1	<b>temp0we.</b> Read-write. Reset: 0. <b>Description:</b> 0: Temp0 is a read-only register, which is the result from Temp0 Sensor. 1: Only Host or EC can write Temp0 register. The Temp0 Sensor result doesn't affect Temp0.
0	<b>inttempwe.</b> Read-write. Reset: 0. <b>Description:</b> 0: IntTemp is a read-only register, which is the result from IntTemp Sensor. 1: Only Host or EC can write IntTemp register. The IntTemp Sensor result doesn't affect IntTemp.

**PM2x00000090 (FCH::PM2::tempstatus)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000090; PM2=FED8\_0400h

Bits	Description
7:6	Reserved.
5	<b>temptsisstatus.</b> Read-write. Reset: 0. Indicate whether TempTsi is out of the limit. Write to 1 to clear.
4	<b>temp3status.</b> Read-write. Reset: 0. Indicate whether Temp3 is out of the limit. Write to 1 to clear.
3	<b>temp2status.</b> Read-write. Reset: 0. Indicate whether Temp2 is out of the limit. Write to 1 to clear.
2	<b>temp1status.</b> Read-write. Reset: 0. Indicate whether Temp1 is out of the limit. Write to 1 to clear.
1	<b>temp0status.</b> Read-write. Reset: 0. Indicate whether Temp0 is out of the limit. Write to 1 to clear.
0	<b>inttempstatus.</b> Read-write. Reset: 0. Indicate whether internal Temp is out of the limit. Write to 1 to clear.

**PM2x00000091 (FCH::PM2::tempcontrol0)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000091; PM2=FED8\_0400h

Bits	Description
7:6	<b>temp2control.</b> Read-write. Reset: 0h. <b>Description:</b> Values other than 00 indicate that Temp2 sensor is enabled. 00: Disable 01: Set Temp2Status to 1 if Temp2 is higher than Temp2Limit 10: Set Temp2Status to 1 if Temp2 is lower than Temp2Limit. 11: Set Temp2Status to 1 if Temp2Hi is higher than Temp2LimitLo (upper bound) or lower than Temp2LimitHi (lower bound).
5:4	<b>temp1control.</b> Read-write. Reset: 0h. <b>Description:</b> Values other than 00 indicate that Temp1 sensor is enabled. 00: Disable 01: Set Temp1Status to 1 if Temp1 is higher than Temp1Limit 10: Set Temp1Status to 1 if Temp1 is lower than Temp1Limit. 11: Set Temp1Status to 1 if Temp1Hi is higher than Temp1LimitLo (upper bound) or lower than Temp1LimitHi (lower bound).
3:2	<b>temp0control.</b> Read-write. Reset: 0h. <b>Description:</b> Values other than 00 indicate that Temp0 sensor is enabled. 00: Disable 01: Set Temp0Status to 1 if Temp0 is higher than Temp0Limit 10: Set Temp0Status to 1 if Temp0 is lower than Temp0Limit. 11: Set Temp0Status to 1 if Temp0Hi is higher than Temp0LimitLo (upper bound) or lower than Temp0LimitHi (lower bound).
1:0	<b>inttempcontrol.</b> Read-write. Reset: 0h. <b>Description:</b> Values other than 00 indicate that IntTemp sensor is enabled. 00: Disable 01: Set IntTempStatus to 1 if IntTemp is higher than IntTempLimit. 10: Set IntTempStatus to 1 if IntTemp is lower than IntTempLimit. 11: Set IntTempStatus to 1 if IntTempHi is higher than IntTempLimitLo or lower than IntTempLimitHi.

**PM2x00000092 (FCH::PM2::tempcontrol1)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000092; PM2=FED8\_0400h

Bits	Description
7:4	Reserved.
3:2	<b>temptsicontrol.</b> Read-write. Reset: 0h. <b>Description:</b> Values other than 00 indicate that TempTsi sensor is enabled. 00: Disable 01: Set TempTsiStatus to 1 if TempTsi is higher than TempTsiLimit 10: Set TempTsiStatus to 1 if TempTsi is lower than TempTsiLimit. 11: Set TempTsiStatus to 1 if TempTsiHi is higher than TempTsiLimitLo (upper bound) or lower than TempTsiLimitHi (lower bound).
1:0	<b>temp3control.</b> Read-write. Reset: 0h. <b>Description:</b> Values other than 00 indicate that Temp3 sensor is enabled. 00: Disable 01: Set Temp3Status to 1 if Temp3 is higher than Temp3Limit 10: Set Temp3Status to 1 if Temp3 is lower than Temp3Limit. 11: Set Temp3Status to 1 if Temp3Hi is higher than Temp3LimitLo (upper bound) or lower than Temp3LimitHi (lower bound).

**PM2x00000093 (FCH::PM2::tempintroute0)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000093; PM2=FED8\_0400h

Bits	Description
7:6	<b>temp2introute</b> . Read-write. Reset: 0h. <b>Description:</b> 01: SMI 10: SMI or SCI according GEVENT 13 INT routing Others: no SCI/SMI generated
5:4	<b>temp1introute</b> . Read-write. Reset: 0h. <b>Description:</b> 01: SMI 10: SMI or SCI according GEVENT 13 INT routing Others: no SCI/SMI generated
3:2	<b>temp0introute</b> . Read-write. Reset: 0h. <b>Description:</b> 01: SMI 10: SMI or SCI according GEVENT 13 INT routing Others: no SCI/SMI generated
1:0	<b>inttempintroute</b> . Read-write. Reset: 0h. <b>Description:</b> 01: SMI 10: SMI or SCI according GEVENT 13 INT routing Others: no SCI/SMI generated

**PM2x00000094 (FCH::PM2::tempintroute1)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000094; PM2=FED8\_0400h

Bits	Description
7:4	Reserved.
3:2	<b>temptsiintroute</b> . Read-write. Reset: 0h. <b>Description:</b> 01: SMI 10: SMI or SCI according GEVENT 13 INT routing Others: no SCI/SMI generated
1:0	<b>temp3introute</b> . Read-write. Reset: 0h. <b>Description:</b> 01: SMI 10: SMI or SCI according GEVENT 13 INT routing Others: no SCI/SMI generated

**PM2x00000095 (FCH::PM2::inttemplo)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x00000095; PM2=FED8\_0400h

Bits	Description
7:0	<b>inttemplo</b> . Read-only. Reset: 00h. IntTemp[7:0]

**PM2x00000096 (FCH::PM2::inttemphi)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x00000096; PM2=FED8\_0400h

Bits	Description
7:0	<b>inttemphi</b> . Read-only. Reset: 00h. IntTemp[15:8]

**PM2x00000097 (FCH::PM2::inttemplimitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000097; PM2=FED8\_0400h

Bits	Description
7:0	<b>inttemplimitlo</b> . Read-write. Reset: 00h. IntTempLimit[7:0]

**PM2x00000098 (FCH::PM2::inttemplimithi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000098; PM2=FED8\_0400h

Bits	Description
7:0	<b>inttemplimithi</b> . Read-write. Reset: 00h. IntTempLimit[15:8]

**PM2x00000099 (FCH::PM2::temp0lo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x00000099; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp0lo</b> . Read-write. Reset: 00h. Temp1[7:0].

**PM2x0000009A (FCH::PM2::temp0hi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000009A; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp0hi</b> . Read-write. Reset: 00h. Temp1[15:8].

**PM2x0000009B (FCH::PM2::temp0limitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000009B; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp0limitlo</b> . Read-write. Reset: 00h. Temp0Limit[7:0]

**PM2x0000009C (FCH::PM2::temp0limithi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000009C; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp0limithi</b> . Read-write. Reset: 00h. Temp0Limit[15:8]

**PM2x0000009D (FCH::PM2::temp1lo)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x0000009D; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp1lo</b> . Read-only. Reset: 00h. Temp1[7:0]

**PM2x0000009E (FCH::PM2::temp1hi)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x0000009E; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp1hi</b> . Read-only. Reset: 00h. Temp1[15:8]

**PM2x0000009F (FCH::PM2::temp1limitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x0000009F; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp1limitlo</b> . Read-write. Reset: 00h. Temp1Limit[7:0]

**PM2x000000A0 (FCH::PM2::temp1limithi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000A0; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp1limithi</b> . Read-write. Reset: 00h. Temp1Limit[15:8]

**PM2x000000A1 (FCH::PM2::temp2lo)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x000000A1; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp2lo</b> . Read-only. Reset: 00h. Temp2[7:0]

**PM2x000000A2 (FCH::PM2::temp2hi)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x000000A2; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp2hi</b> . Read-only. Reset: 00h. Temp2[15:8]

**PM2x000000A3 (FCH::PM2::temp2limitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000A3; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp2limitlo</b> . Read-write. Reset: 00h. Temp2Limit[7:0]

**PM2x000000A4 (FCH::PM2::temp2limithi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000A4; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp2limithi</b> . Read-write. Reset: 00h. Temp2Limit[15:8]

**PM2x000000A5 (FCH::PM2::temp3lo)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x000000A5; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp3lo</b> . Read-only. Reset: 00h. Temp3[7:0]

**PM2x000000A6 (FCH::PM2::temp3hi)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x000000A6; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp3hi</b> . Read-only. Reset: 00h. Temp3[15:8]

**PM2x000000A7 (FCH::PM2::temp3limitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000A7; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp3limitlo</b> . Read-write. Reset: 00h. Temp3Limit[7:0]

**PM2x000000A8 (FCH::PM2::temp3limithi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000A8; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp3limithi</b> . Read-write. Reset: 00h. Temp3Limit[15:8]

**PM2x000000AB (FCH::PM2::inttempchangelimit)**

Read-write. Reset: 00h.

When  $\text{Temp}(\text{new}) > \text{Temp}(\text{old}) + [\text{TempChangeLimit} \ll 6]$  $\text{Temp} = \text{Temp}(\text{old}) + [\text{TempChangeLimit} \ll 6]$ When  $\text{Temp}(\text{new}) < \text{Temp}(\text{old}) [\text{TempChangeLimit} \ll 6]$  $\text{Temp} = \text{Temp}(\text{old}) [\text{TempChangeLimit} \ll 6]$ When  $\text{Temp}(\text{new}) \leq \text{Temp}(\text{old}) + [\text{TempChangeLimit} \ll 6]$  and  $\geq \text{Temp}(\text{old}) [\text{TempChangeLimit} \ll 6]$  $\text{Temp} = \text{Temp}(\text{new})$ \_aliasHOST; PM2x000000AB; PM2=FED8\_0400h

Bits	Description
7:0	<b>tempchangelimit</b> . Read-write. Reset: 00h. Filtering is applied to {IntTempHi, IntTempLo} if TempChangeLimit is nonzero.

**PM2x000000AC (FCH::PM2::temp0changelimit)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000AC; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp0changelimit</b> . Read-write. Reset: 00h. Filtering is applied to {Temp0Hi, Temp0Lo} if TempChangeLimit is nonzero.

**PM2x000000AD (FCH::PM2::temp1changelimit)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000AD; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp1changelimit</b> . Read-write. Reset: 00h. Filtering is applied to {Temp1Hi, Temp1Lo} if TempChangeLimit is nonzero.

**PM2x000000AE (FCH::PM2::temp2changelimit)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000AE; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp2changelimit</b> . Read-write. Reset: 00h. Filtering is applied to {Temp2Hi, Temp2Lo} if TempChangeLimit is nonzero.

**PM2x000000AF (FCH::PM2::temp3changelimit)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000AF; PM2=FED8\_0400h

Bits	Description
7:0	<b>temp3changelimit</b> . Read-write. Reset: 00h. Filtering is applied to {Temp3Hi, Temp3Lo} if TempChangeLimit is nonzero.

**PM2x000000B0 (FCH::PM2::voltagestatus)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x000000B0; PM2=FED8\_0400h

Bits	Description
7	<b>voltage7status.</b> Read-only. Reset: 0. 1 means that Vin7 is out of VoltageLimit7
6	<b>voltage6status.</b> Read-only. Reset: 0. 1 means that Vin6 is out of VoltageLimit6
5	<b>voltage5status.</b> Read-only. Reset: 0. 1 means that Vin5 is out of VoltageLimit5
4	<b>voltage4status.</b> Read-only. Reset: 0. 1 means that Vin4 is out of VoltageLimit4
3	<b>voltage3status.</b> Read-only. Reset: 0. 1 means that Vin3 is out of VoltageLimit3
2	<b>voltage2status.</b> Read-only. Reset: 0. 1 means that Vin2 is out of VoltageLimit2
1	<b>voltage1status.</b> Read-only. Reset: 0. 1 means that Vin1 is out of VoltageLimit1
0	<b>voltage0status.</b> Read-only. Reset: 0. 1 means that Vin0 is out of VoltageLimit0



**PM2x000000B2 (FCH::PM2::voltagecontrol0)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000B2; PM2=FED8\_0400h

Bits	Description
7:6	<b>voltage3control.</b> Read-write. Reset: 0h. <b>Description:</b> Values other than 00 indicate that Voltage3 sensor is enabled. 00: Disable 01: Monitor current value (combined value from registers 66h and 65h) against Voltage3Limit (combined value from registers 68h and 67h) set Voltage3Status bit if is greater than Voltage3Limit 10: Monitor current value (combined value from registers 66h and 65h) against Voltage3Limit (combined value from registers 68h and 67h) set Voltage3Status bit if it is lower than Voltage3Limit 11: Monitor Voltage3Hi against the limits. Set Voltage3Status when it is more than Voltage3LimitLo [7:0] or lower than Voltage3LimitHi [15:8]
5:4	<b>voltage2control.</b> Read-write. Reset: 0h. <b>Description:</b> Values other than 00 indicate that Voltage2 sensor is enabled. 00: Disable 01: Monitor current value (combined value from registers 62h and 61h) against Voltage2Limit (combined value from registers 64h and 63h) set Voltage2Status bit if is greater than Voltage2Limit 10: Monitor current value (combined value from registers 62h and 61h) against Voltage2Limit (combined value from registers 64h and 63h) set Voltage2Status bit if it is lower than Voltage2Limit 11: Monitor Voltage2Hi against the limits. Set Voltage2Status when it is more than Voltage2LimitLo [7:0] or Lower than Voltage2LimitHi [15:8]
3:2	<b>voltage1control.</b> Read-write. Reset: 0h. <b>Description:</b> Values other than 00 indicate that Voltage1 sensor is enabled. 00: Disable 01: Monitor current value (combined value from registers 5Eh and 5Dh) against Voltage1Limit (combined value from registers 60h and 5Fh) set Voltage1Status bit if is greater than Voltage1Limit 10: Monitor current value (combined value from registers 5Eh and 5Dh) against Voltage1Limit (combined value from registers 60h and 5Fh) set Voltage1Status bit if it is lower than Voltage1Limit 11: Monitor Voltage1Hi against the limits. Set Voltage1 Status when it is more than Voltage1LimitLo [7:0] or Lower than Voltage1LimitHi [15:8]
1:0	<b>voltage0control.</b> Read-write. Reset: 0h. <b>Description:</b> Values other than 00 indicate that Voltage0 sensor is enabled. 00: Disable 01: Monitor current value (combined value from registers 5Ah and 59h) against Voltage0Limit (combined value from registers 5Ch and 5Bh) set Voltage0Status bit if is greater than Voltage0Limit 10: Monitor current value (combined value from registers 5Ah and 59h) against Voltage0Limit (combined value from registers 5Ch and 5Bh) set Voltage0Status bit if it is lower than Voltage0Limit 11: Monitor Voltage0Hi against the limits. Set Voltage0Status when it is more than Voltage0LimitLo [7:0] or Lower than Voltage0LimitHi [15:8]

**PM2x000000B3 (FCH::PM2::voltagecontrol1)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000B3; PM2=FED8\_0400h

Bits	Description
7:6	<b>voltage7control.</b> Read-write. Reset: 0h. <b>Description:</b> Values other than 00 indicate that Voltage7 sensor is enabled. 00: Disable 01: Monitor current value (combined value from registers 76h and 75h) against Voltage7Limit (combined value from registers 78h and 77h) set Voltage7Status bit if is greater than Voltage7Limit 10: Monitor current value (combined value from registers 76h and 75h) against Voltage7Limit (combined value from registers 78h and 77h) set Voltage7Status bit if it is lower than Voltage7Limit 11: Monitor Voltage7Hi against the limits. Set Voltage7Status when it is more than Voltage7LimitLo [7:0] or lower than Voltage7LimitHi [15:8]
5:4	<b>voltage6control.</b> Read-write. Reset: 0h. <b>Description:</b> Values other than 00 indicate that Voltage6 sensor is enabled. 00: Disable 01: Monitor current value (combined value from registers 72h and 71h) against Voltage6Limit (combined value from registers 74h and 73h) set Voltage6Status bit if is greater than Voltage6Limit 10: Monitor current value (combined value from registers 72h and 71h) against Voltage6Limit (combined value from registers 74h and 73h) set Voltage6Status bit if it is lower than Voltage6Limit 11: Monitor Voltage6Hi against the limits. Set Voltage6Status when it is more than Voltage6LimitLo [7:0] or lower than Voltage6LimitHi [15:8]
3:2	<b>voltage5control.</b> Read-write. Reset: 0h. <b>Description:</b> Values other than 00 indicate that Voltage5 sensor is enabled. 00: Disable 01: Monitor current value (combined value from registers 6Eh and 6Dh) against Voltage5Limit (combined value from registers 70h and 6Fh) set Voltage5Status bit if is greater than Voltage5Limit 10: Monitor current value (combined value from registers 6Eh and 6Dh) against Voltage5Limit (combined value from registers 70h and 6Fh) set Voltage5Status bit if it is lower than Voltage5Limit 11: Monitor Voltage5Hi against the limits. Set Voltage5Status when it is more than Voltage5LimitLo [7:0] or lower than Voltage5LimitHi [15:8]
1:0	<b>voltage4control.</b> Read-write. Reset: 0h. <b>Description:</b> Values other than 00 indicate that Voltage4 sensor is enabled. 00: Disable 01: Monitor current value (combined value from registers 6Ah and 69h) against Voltage4Limit (combined value from registers 6Ch and 6Bh) set Voltage4Status bit if is greater than Voltage4Limit 10: Monitor current value (combined value from registers 6Ah and 69h) against Voltage4Limit (combined value from registers 6Ch and 6Bh) set Voltage4Status bit if it is lower than Voltage4Limit 11: Monitor Voltage4Hi against the limits. Set Voltage4Status when it is more than Voltage4LimitLo [7:0] or lower than Voltage4LimitHi [15:8]

**PM2x000000B5 (FCH::PM2::analogintroute0)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000B5; PM2=FED8\_0400h

Bits	Description
7:6	<b>voltage3introute.</b> Read-write. Reset: 0h. <b>Description:</b> 01: SMI 10: SMI or SCI according GEVENT 13 INT routing Others: no SCI/SMI generated
5:4	<b>voltage2introute.</b> Read-write. Reset: 0h. <b>Description:</b> 01: SMI 10: SMI or SCI according GEVENT 13 INT routing Others: no SCI/SMI generated
3:2	<b>voltage1introute.</b> Read-write. Reset: 0h. <b>Description:</b> 01: SMI 10: SMI or SCI according GEVENT 13 INT routing Others: no SCI/SMI generated
1:0	<b>voltage0introute.</b> Read-write. Reset: 0h. <b>Description:</b> 01: SMI 10: SMI or SCI according GEVENT 13 INT routing Others: no SCI/SMI generated

**PM2x000000B6 (FCH::PM2::analogintroute1)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000B6; PM2=FED8\_0400h

Bits	Description
7:6	<b>voltage7introute.</b> Read-write. Reset: 0h. <b>Description:</b> 01: SMI 10: SMI or SCI according GEVENT 13 INT routing Others: no SCI/SMI generated
5:4	<b>voltage6introute.</b> Read-write. Reset: 0h. <b>Description:</b> 01: SMI 10: SMI or SCI according GEVENT 13 INT routing Others: no SCI/SMI generated
3:2	<b>voltage5introute.</b> Read-write. Reset: 0h. <b>Description:</b> 01: SMI 10: SMI or SCI according GEVENT 13 INT routing Others: no SCI/SMI generated
1:0	<b>voltage4introute.</b> Read-write. Reset: 0h. <b>Description:</b> 01: SMI 10: SMI or SCI according GEVENT 13 INT routing Others: no SCI/SMI generated

**PM2x000000B8 (FCH::PM2::voltage0lo)**

Read-only. Reset: 00h.

Voltage0Lo and Voltage0Hi returns the read value from VIN0 input.

\_aliasHOST; PM2x000000B8; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage0lo.</b> Read-only. Reset: 00h. Voltage0 [7:0]

**PM2x000000B9 (FCH::PM2::voltage0hi)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x000000B9; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage0hi</b> . Read-only. Reset: 00h. Voltage0 [15:8]

**PM2x000000BA (FCH::PM2::voltage0limitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000BA; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage0limitlo</b> . Read-write. Reset: 00h. Voltage0Limit[7:0]

**PM2x000000BB (FCH::PM2::voltage0limithi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000BB; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage0limithi</b> . Read-write. Reset: 00h. Voltage0Limit[15:8]

**PM2x000000BC (FCH::PM2::voltage1lo)**

Read-only. Reset: 00h.

Voltage0Lo and Voltage0Hi returns the read value from VIN0 input.

\_aliasHOST; PM2x000000BC; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage1lo</b> . Read-only. Reset: 00h. Voltage1 [7:0]

**PM2x000000BD (FCH::PM2::voltage1hi)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x000000BD; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage1hi</b> . Read-only. Reset: 00h. Voltage1 [15:8]

**PM2x000000BE (FCH::PM2::voltage1limitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000BE; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage1limitlo</b> . Read-write. Reset: 00h. Voltage1Limit[7:0]

**PM2x000000BF (FCH::PM2::voltage1limithi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000BF; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage1limithi</b> . Read-write. Reset: 00h. Voltage1Limit[15:8]

**PM2x000000C0 (FCH::PM2::voltage2lo)**

Read-only. Reset: 00h.

Voltage0Lo and Voltage0Hi returns the read value from VIN0 input.

\_aliasHOST; PM2x000000C0; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage2lo</b> . Read-only. Reset: 00h. Voltage2[7:0]

**PM2x000000C1 (FCH::PM2::voltage2hi)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x000000C1; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage2hi</b> . Read-only. Reset: 00h. Voltage2[15:8]

**PM2x000000C2 (FCH::PM2::voltage2limitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000C2; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage2limitlo</b> . Read-write. Reset: 00h. Voltage2Limit[7:0]

**PM2x000000C3 (FCH::PM2::voltage2limithi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000C3; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage2limithi</b> . Read-write. Reset: 00h. Voltage2Limit[15:8]

**PM2x000000C4 (FCH::PM2::voltage3lo)**

Read-only. Reset: 00h.

Voltage0Lo and Voltage0Hi returns the read value from VIN0 input.

\_aliasHOST; PM2x000000C4; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage3lo</b> . Read-only. Reset: 00h. Voltage3[7:0]

**PM2x000000C5 (FCH::PM2::voltage3hi)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x000000C5; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage3hi</b> . Read-only. Reset: 00h. Voltage3[15:8]

**PM2x000000C6 (FCH::PM2::voltage3limitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000C6; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage3limitlo</b> . Read-write. Reset: 00h. Voltage3Limit[7:0]

**PM2x000000C7 (FCH::PM2::voltage3limithi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000C7; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage3limithi</b> . Read-write. Reset: 00h. Voltage3Limit[15:8]

**PM2x000000C8 (FCH::PM2::voltage4lo)**

Read-only. Reset: 00h.

Voltage0Lo and Voltage0Hi returns the read value from VIN0 input.

\_aliasHOST; PM2x000000C8; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage4lo</b> . Read-only. Reset: 00h. Voltage4[7:0]

**PM2x000000C9 (FCH::PM2::voltage4hi)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x000000C9; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage4hi</b> . Read-only. Reset: 00h. Voltage4[15:8]

**PM2x000000CA (FCH::PM2::voltage4limitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000CA; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage4limitlo</b> . Read-write. Reset: 00h. Voltage4Limit[7:0]

**PM2x000000CB (FCH::PM2::voltage4limithi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000CB; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage4limithi</b> . Read-write. Reset: 00h. Voltage4Limit[15:8]

**PM2x000000CC (FCH::PM2::voltage5lo)**

Read-only. Reset: 00h.

Voltage0Lo and Voltage0Hi returns the read value from VIN0 input.

\_aliasHOST; PM2x000000CC; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage5lo</b> . Read-only. Reset: 00h. Voltage5[7:0]

**PM2x000000CD (FCH::PM2::voltage5hi)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x000000CD; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage5hi</b> . Read-only. Reset: 00h. Voltage5[15:8]

**PM2x000000CE (FCH::PM2::voltage5limitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000CE; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage5limitlo</b> . Read-write. Reset: 00h. Voltage5Limit[7:0]

**PM2x000000CF (FCH::PM2::voltage5limithi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000CF; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage5limithi</b> . Read-write. Reset: 00h. Voltage5Limit[15:8]

**PM2x000000D0 (FCH::PM2::voltage6lo)**

Read-only. Reset: 00h.

Voltage0Lo and Voltage0Hi returns the read value from VIN0 input.

\_aliasHOST; PM2x000000D0; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage6lo</b> . Read-only. Reset: 00h. Voltage6[7:0]

**PM2x000000D1 (FCH::PM2::voltage6hi)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x000000D1; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage6hi</b> . Read-only. Reset: 00h. Voltage6[15:8]

**PM2x000000D2 (FCH::PM2::voltage6limitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000D2; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage6limitlo</b> . Read-write. Reset: 00h. Voltage6Limit[7:0]

**PM2x000000D3 (FCH::PM2::voltage6limithi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000D3; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage6limithi</b> . Read-write. Reset: 00h. Voltage6Limit[15:8]

**PM2x000000D4 (FCH::PM2::voltage7lo)**

Read-only. Reset: 00h.

Voltage0Lo and Voltage0Hi returns the read value from VIN0 input.

\_aliasHOST; PM2x000000D4; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage7lo</b> . Read-only. Reset: 00h. Voltage7[7:0]

**PM2x000000D5 (FCH::PM2::voltage7hi)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x000000D5; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage7hi</b> . Read-only. Reset: 00h. Voltage7[15:8]

**PM2x000000D6 (FCH::PM2::voltage7limitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000D6; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage7limitlo</b> . Read-write. Reset: 00h. Voltage7Limit[7:0]

**PM2x000000D7 (FCH::PM2::voltage7limithi)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000D7; PM2=FED8\_0400h

Bits	Description
7:0	<b>voltage7limithi</b> . Read-write. Reset: 00h. Voltage7Limit[15:8]

**PM2x000000DD (FCH::PM2::analogiorstsel)**

Read-write. Reset: 00h.

The HWM can be enabled or disabled during S3 or S5 state, depends on the setting of HWM\_ResetOption (PMIO\_Reg:BE[6]). If it is enabled (HWM\_ResetOption = 0) during S3 or S5 states, the PMIO2\_Reg:DD can be used to mask off the events/status caused by voltage sensors.

\_aliasHOST; PM2x000000DD; PM2=FED8\_0400h

Bits	Description
7	<b>voltage7rstsel.</b> Read-write. Reset: 0. <b>Description:</b> If set, VIN7 (voltage monitoring) function will not be stopped by PCI reset. 0: 483riting 1: enable
6	<b>voltage6rstsel.</b> Read-write. Reset: 0. <b>Description:</b> If set, VIN6 (voltage monitoring) function will not be stopped by PCI reset. 0: 483riting 1: enable
5	<b>voltage5rstsel.</b> Read-write. Reset: 0. <b>Description:</b> If set, VIN5 (voltage monitoring) function will not be stopped by PCI reset. 0: 483riting 1: enable
4	<b>voltage4rstsel.</b> Read-write. Reset: 0. <b>Description:</b> If set, VIN4 (voltage monitoring) function will not be stopped by PCI reset. 0: 483riting 1: enable
3	<b>voltage3rstsel.</b> Read-write. Reset: 0. <b>Description:</b> If set, VIN3 (voltage monitoring) function will not be stopped by PCI reset. 0: 482riting 1: enable
2	<b>voltage2rstsel.</b> Read-write. Reset: 0. <b>Description:</b> If set, VIN2 (voltage monitoring) function will not be stopped by PCI reset. 0: 482riting 1: enable
1	<b>voltage1rstsel.</b> Read-write. Reset: 0. <b>Description:</b> If set, VIN1 (voltage monitoring) function will not be stopped by PCI reset. 0: 482riting 1: enable
0	<b>voltage0rstsel.</b> Read-write. Reset: 0. <b>Description:</b> If set, VIN0 (voltage monitoring) function will not be stopped by PCI reset. 0: 482riting 1: enable



**PM2x000000DF (FCH::PM2::temprstsel)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000DF; PM2=FED8\_0400h

Bits	Description
7:6	Reserved.
5	<b>temptstsel.</b> Read-write. Reset: 0. <b>Description:</b> If set, TEMPIN4 (thermal diode monitoring) function will not be stopped by reset. 0: 486riting 1: enable
4	<b>temp3rstsel.</b> Read-write. Reset: 0. <b>Description:</b> If set, TEMPIN3 (thermal diode monitoring) function will not be stopped by reset. 0: 486riting 1: enable
3	<b>temp2rstsel.</b> Read-write. Reset: 0. <b>Description:</b> If set, TEMPIN2 (thermal diode monitoring) function will not be stopped by reset. 0: 485riting 1: enable
2	<b>temp1rstsel.</b> Read-write. Reset: 0. <b>Description:</b> If set, TEMPIN1 (thermal diode monitoring) function will not be stopped by reset. 0: 485riting 1: enable
1	<b>temp0rstsel.</b> Read-write. Reset: 0. <b>Description:</b> If set, TEMPIN0 (thermal diode monitoring) function will not be stopped by reset. 0: 485riting 1: enable
0	<b>inttemprstsel.</b> Read-write. Reset: 0. <b>Description:</b> If set, the internal diode temperature sensing function will not be stopped by reset. 0: 484riting 1: enable

**PM2x000000E0 (FCH::PM2::alertthermaltripstatus)**

Read-only.

\_aliasHOST; PM2x000000E0; PM2=FED8\_0400h

Bits	Description
7:2	Reserved.
1	<b>thermaltripstatus.</b> Read-only. Reset: 0. <b>Description:</b> Read only. 0: Current temperature is not above ThermalTripLimit 1: Current temperature is above ThermalTripLimit
0	<b>alertstatus.</b> Read-only. Reset: 0. <b>Description:</b> Read only. 0: Current temperature is not above AlertLimit 1: Current temperature is above AlertLimit

**PM2x000000E1 (FCH::PM2::alertlimitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000E1; PM2=FED8\_0400h

Bits	Description
7:0	<b>alertlimit.</b> Read-write. Reset: 00h. AlertLimit[7:0]

**PM2x000000E2 (FCH::PM2::alertlimiti)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000E2; PM2=FED8\_0400h

Bits	Description
7:0	<b>alertlimit</b> . Read-write. Reset: 00h. AlertLimit[15:8]

**PM2x000000E3 (FCH::PM2::thermaltriplimitlo)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000E3; PM2=FED8\_0400h

Bits	Description
7:0	<b>thermaltriplimit</b> . Read-write. Reset: 00h. ThermalTripLimit [7:0]

**PM2x000000E4 (FCH::PM2::thermaltriplimiti)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000E4; PM2=FED8\_0400h

Bits	Description
7:0	<b>thermaltriplimit</b> . Read-write. Reset: 00h. ThermalTripLimit [15:8]

**PM2x000000E5 (FCH::PM2::alertthermaltripcontrol)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000E5; PM2=FED8\_0400h

Bits	Description
7:5	<b>tempselalert</b> . Read-write. Reset: 0h. <b>Description:</b> Select temperature sensor as event source 000: IntTemp 001: Temp0 010: Temp1 011: Temp2 100: Temp3 101: TempTsi Others: Function disabled This register converts the Temp* pin into either Talert or ThermalTrip function.
4:2	Reserved.
1:0	<b>alertcontrol</b> . Read-write. Reset: 0h. <b>Description:</b> Bit 0: Enable Talert on the selected Temp input Bit 1: Enable ThermalTrip on the selected Temp input

**PM2x000000E6 (FCH::PM2::hwmcontrol)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000E6; PM2=FED8\_0400h

Bits	Description
7	<b>resetsensor</b> . Read-write. Reset: 0. <b>Description:</b> 0: Running state 1: Put the HWM into reset state.
6:4	Reserved.
3	<b>pdalways</b> . Read-write. Reset: 0. <b>Description:</b> 0: Power on the HWM only when doing a sensor reading. 1: Power on the HWM all the time.
2	<b>fastreadenable</b> . Read-write. Reset: 0. Set to 1 to keep HWM reading sensors repeatedly.
1	<b>autoreadsensor</b> . Read-write. Reset: 0. Set to 1 to enable periodical reading of voltage/temperature sensors
0	<b>hostreadsensor</b> . Read-write. Reset: 0. Writing to 1 forces HWM to do a read.

**PM2x000000E7 (FCH::PM2::voltage\_readfreq)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000E7; PM2=FED8\_0400h

Bits	Description
7	<b>voltreadus.</b> Read-write. Reset: 0. <b>Description:</b> 1: Voltage read period in s 0: Voltage read period in ms
6:2	Reserved.
1:0	<b>voltsensorreadfreq.</b> Read-write. Reset: 0h. <b>Description:</b> 00: 100 Hz 01: 200 Hz 10: 300 Hz 11: 500 Hz

**PM2x000000E8 (FCH::PM2::temp\_readfreq)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000E8; PM2=FED8\_0400h

Bits	Description
7:2	Reserved.
1:0	<b>tempsensorreadfreq.</b> Read-write. Reset: 0h. <b>Description:</b> 00: 100ms 01: 200ms 10: 300ms 11: 500ms

**PM2x000000E9 (FCH::PM2::voltage\_readaverage)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000E9; PM2=FED8\_0400h

Bits	Description
7	<b>volateg7_average.</b> Read-write. Reset: 0. Set to 1 to enable cumulative averaging of Vin7.
6	<b>volateg6_average.</b> Read-write. Reset: 0. Set to 1 to enable cumulative averaging of Vin6.
5	<b>volateg5_average.</b> Read-write. Reset: 0. Set to 1 to enable cumulative averaging of Vin5.
4	<b>volateg4_average.</b> Read-write. Reset: 0. Set to 1 to enable cumulative averaging of Vin4.
3	<b>volateg3_average.</b> Read-write. Reset: 0. Set to 1 to enable cumulative averaging of Vin3.
2	<b>volateg2_average.</b> Read-write. Reset: 0. Set to 1 to enable cumulative averaging of Vin2.
1	<b>volateg1_average.</b> Read-write. Reset: 0. Set to 1 to enable cumulative averaging of Vin1.
0	<b>volateg0_average.</b> Read-write. Reset: 0. Set to 1 to enable cumulative averaging of Vin0.

**PM2x000000EA (FCH::PM2::hwm\_voltcalib)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x000000EA; PM2=FED8\_0400h

Bits	Description
7	<b>voltage_calibrationvalid.</b> Read-only. Reset: 0. <b>Description:</b> 1: Voltage_calibration in this register is valid. 0: Voltage_calibration in this register is invalid.
6:0	<b>voltage_calibration.</b> Read-only. Reset: 00h. Hwm calibration value

**PM2x000000EB (FCH::PM2::tempreadaverage)**

Read-write. Reset: 39h.

\_aliasHOST; PM2x000000EB; PM2=FED8\_0400h

Bits	Description
7:6	Reserved.
5	<b>temptsi_average</b> . Read-write. Reset: 1. Set to 1 to enable cumulative averaging of TempTsi.
4	<b>temp3_average</b> . Read-write. Reset: 1. Set to 1 to enable cumulative averaging of Temp3.
3	<b>temp2_average</b> . Read-write. Reset: 1. Set to 1 to enable cumulative averaging of Temp2.
2	<b>temp1_average</b> . Read-write. Reset: 0. Set to 1 to enable cumulative averaging of Temp1.
1	<b>temp0_average</b> . Read-write. Reset: 0. Set to 1 to enable cumulative averaging of Temp0.
0	<b>inttemp_average</b> . Read-write. Reset: 1. Set to 1 to enable cumulative averaging of Internal Temp.

**PM2x000000EC (FCH::PM2::hwmstatus)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x000000EC; PM2=FED8\_0400h

Bits	Description
7	<b>hostreadsts</b> . Read-only. Reset: 0. <b>Description:</b> 0: No host read is pending. 1: Host read is pending.
6:1	Reserved.
0	<b>sensoridle</b> . Read-only. Reset: 0. <b>Description:</b> 0: HWM is idle. 1: HWM is doing the sensor reading.

**PM2x000000ED (FCH::PM2::voltage7readstatus)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x000000ED; PM2=FED8\_0400h

Bits	Description
7	<b>voltage7readstatus.</b> Read-only. Reset: 0. <b>Description:</b> 0: No Voltage7 reading is pending. 1: Voltage7 reading is pending.
6	<b>voltage6readstatus.</b> Read-only. Reset: 0. <b>Description:</b> 0: No Voltage6 reading is pending. 1: Voltage6 reading is pending.
5	<b>voltage5readstatus.</b> Read-only. Reset: 0. <b>Description:</b> 0: No Voltage5 reading is pending. 1: Voltage5 reading is pending.
4	<b>voltage4readstatus.</b> Read-only. Reset: 0. <b>Description:</b> 0: No Voltage4 reading is pending. 1: Voltage4 reading is pending.
3	<b>voltage3readstatus.</b> Read-only. Reset: 0. <b>Description:</b> 0: No Voltage3 reading is pending. 1: Voltage3 reading is pending.
2	<b>voltage2readstatus.</b> Read-only. Reset: 0. <b>Description:</b> 0: No Voltage2 reading is pending. 1: Voltage2 reading is pending.
1	<b>voltage1readstatus.</b> Read-only. Reset: 0. <b>Description:</b> 0: No Voltage1 reading is pending. 1: Voltage1 reading is pending.
0	<b>voltage0readstatus.</b> Read-only. Reset: 0. <b>Description:</b> 0: No Voltage0 reading is pending. 1: Voltage0 reading is pending.

**PM2x000000EE (FCH::PM2::tempreadstatus)**

Read-only. Reset: 00h.

\_aliasHOST; PM2x000000EE; PM2=FED8\_0400h

Bits	Description
7:5	Reserved.
4	<b>temp3readstatus.</b> Read-only. Reset: 0. <b>Description:</b> 0: No Temp3 reading is pending. 1: Temp3 reading is pending.
3	<b>temp2readstatus.</b> Read-only. Reset: 0. <b>Description:</b> 0: No Temp2 reading is pending. 1: Temp2 reading is pending.
2	<b>temp1readstatus.</b> Read-only. Reset: 0. <b>Description:</b> 0: No Temp1 reading is pending. 1: Temp1 reading is pending.
1	<b>temp0readstatus.</b> Read-only. Reset: 0. <b>Description:</b> 0: No Temp0 reading is pending. 1: Temp0 reading is pending.
0	<b>inttempreadstatus.</b> Read-only. Reset: 0. <b>Description:</b> 0: No Internal Temp reading is pending. 1: Internal Temp reading is pending.

**PM2x000000EF (FCH::PM2::hwmclkcontrol)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000EF; PM2=FED8\_0400h

Bits	Description
7:4	Reserved.
3:0	<b>sensorclkdiv.</b> Read-write. Reset: 0h. <b>Description:</b> To set the Hwm_Clk (sampling clock rate) frequency. Hwm_Clk = 66.67MHz / (2 * (SensorClkDiv+2)) For example, Hwm_Clk = 16.67MHz when the SensorClkDiv is set to value of 0000 Hwm_Clk = 11.11MHz when the SensorClkDiv is set to value of 0001

**PM2x000000F0 (FCH::PM2::adc\_pdbtime)**

Read-write. Reset: 19h.

\_aliasHOST; PM2x000000F0; PM2=FED8\_0400h

Bits	Description
7:5	Reserved.
4:0	<b>adc_pdbtime.</b> Read-write. Reset: 19h. Control the time between HWM_PDB and ADC_PDB. Its unit is s. Default is 25 s.

**PM2x000000F1 (FCH::PM2::adc\_startup)**

Read-write. Reset: 0Ah.

\_aliasHOST; PM2x000000F1; PM2=FED8\_0400h

Bits	Description
7:4	Reserved.
3:0	<b>adc_startup.</b> Read-write. Reset: Ah. Control the time between ADC_RESET and sensor reading. Its unit is s. Default is 10 s.

**PM2x000000F2 (FCH::PM2::adc\_delay)**

Read-write. Reset: 02h.

\_aliasHOST; PM2x000000F2; PM2=FED8\_0400h

Bits	Description
7:4	Reserved.
3:0	<b>adc_startup.</b> Read-write. Reset: 2h. Control the delay time between two back to back reading. Its unit is us. Default is 2 s.

**PM2x000000F3 (FCH::PM2::sax\_ctl\_vtime)**

Read-write. Reset: 05h.

\_aliasHOST; PM2x000000F3; PM2=FED8\_0400h

Bits	Description
7:4	Reserved.
3:0	<b>sax_ctl_vtime.</b> Read-write. Reset: 5h. Control the assertion time of SAX_CTL_V. Default is 5 s.

**PM2x000000F4 (FCH::PM2::sax\_ctl\_ttime)**

Read-write. Reset: C8h.

\_aliasHOST; PM2x000000F4; PM2=FED8\_0400h

Bits	Description
7:0	<b>sax_ctl_ttime.</b> Read-write. Reset: C8h. Control the assertion time of SAX_CTL_T. Default is 200 s.

**PM2x000000F5 (FCH::PM2::bgadj)**

Read-write. Reset: 20h.

\_aliasHOST; PM2x000000F5; PM2=FED8\_0400h

Bits	Description
7:6	Reserved.
5:0	<b>bgadj</b> . Read-write. Reset: 20h. HWM tuning parameter

**PM2x000000F6 (FCH::PM2::afecfg\_clkdiv)**

Read-write. Reset: 80h.

\_aliasHOST; PM2x000000F6; PM2=FED8\_0400h

Bits	Description
7:6	<b>clk_div</b> . Read-write. Reset: 2h. HWM tuning parameter
5:2	Reserved.
1:0	<b>afe_cfg</b> . Read-write. Reset: 0h. HWM tuning parameter

**PM2x000000F7 (FCH::PM2::hwm\_debugsel)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000F7; PM2=FED8\_0400h

Bits	Description
7:4	Reserved.
3:0	<b>hwm_macro_debugsel</b> . Read-write. Reset: 0h. HWM tuning parameter

**PM2x000000F8 (FCH::PM2::voltagesamplesel)**

Read-write. Reset: 04h.

\_aliasHOST; PM2x000000F8; PM2=FED8\_0400h

Bits	Description
7	<b>hiratioenablefortemp3</b> . Read-write. Reset: 0. Set to 1 to enable hi current ratioc on Temp3.
6	<b>hiratioenablefortemp2</b> . Read-write. Reset: 0. Set to 1 to enable hi current ratioc on Temp2.
5	<b>hiratioenablefortemp1</b> . Read-write. Reset: 0. Set to 1 to enable hi current ratioc on Temp1.
4	<b>hiratioenablefortemp0</b> . Read-write. Reset: 0. Set to 1 to enable hi current ratioc on Temp0.
3	<b>hiratioenableforinttemp</b> . Read-write. Reset: 0. Set to 1 to enable hi current ratioc on Internal Temp.
2:0	<b>num_samples_forvolt</b> . Read-write. Reset: 4h. <b>Description:</b> Specify number of samples per voltage reading. Default value is 1 sample per reading. 000 : 1 sample (default) 001 : 2 samples 010 : 4 samples 011 : 8 samples 100 : 16 samples 101 : 32 samples 110 : 64 samples

**PM2x000000F9 (FCH::PM2::tempsamplesel)**

Read-write. Reset: 06h.

\_aliasHOST; PM2x000000F9; PM2=FED8\_0400h

Bits	Description
7	<b>hicurenalefortemp3</b> . Read-write. Reset: 0. Set to 1 to enable hi current on temp3.
6	<b>hicurenalefortemp2</b> . Read-write. Reset: 0. Set to 1 to enable hi current on temp2.
5	<b>hicurenalefortemp1</b> . Read-write. Reset: 0. Set to 1 to enable hi current on temp1.
4	<b>hicurenalefortemp0</b> . Read-write. Reset: 0. Set to 1 to enable hi current on temp0.
3	<b>hicurenaleforinttemp</b> . Read-write. Reset: 0. Set to 1 to enable hi current on internal temp.
2:0	<b>num_samples_fortemp</b> . Read-write. Reset: 6h. <b>Description:</b> Specify number of samples per Temp reading. Default value is 16 samples per reading 000 : 1 sample 001 : 2 samples 010 : 4 samples 011 : 8 samples 100 : 16 samples (default) 101 : 32 samples 110 : 64 samples

**PM2x000000FA (FCH::PM2::hwmvoltage\_div0)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000FA; PM2=FED8\_0400h

Bits	Description
7:6	<b>volt3_div</b> . Read-write. Reset: 0h. Specify voltage3 read range.
5:4	<b>volt2_div</b> . Read-write. Reset: 0h. Specify voltage2 read range.
3:2	<b>volt1_div</b> . Read-write. Reset: 0h. Specify voltage1 read range.
1:0	<b>volt0_div</b> . Read-write. Reset: 0h. Specify voltage0 read range.

**PM2x000000FB (FCH::PM2::hwmvoltage\_div1)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000FB; PM2=FED8\_0400h

Bits	Description
7:6	<b>volt7_div</b> . Read-write. Reset: 0h. Specify voltage7 read range.
5:4	<b>volt6_div</b> . Read-write. Reset: 0h. Specify voltage6 read range.
3:2	<b>volt5_div</b> . Read-write. Reset: 0h. Specify voltage5 read range.
1:0	<b>volt4_div</b> . Read-write. Reset: 0h. Specify voltage4 read range.

**PM2x000000FC (FCH::PM2::adc\_gain\_adj)**

Read-write. Reset: 08h.

\_aliasHOST; PM2x000000FC; PM2=FED8\_0400h

Bits	Description
7:4	Reserved.
3:0	<b>adc_gain_adj</b> . Read-write. Reset: 8h. HWM tuning parameter.

**PM2x000000FD (FCH::PM2::adc\_cfg)**

Read-write. Reset: 08h.

\_aliasHOST; PM2x000000FD; PM2=FED8\_0400h

Bits	Description
7:4	Reserved.
3:0	<b>adc_cfg</b> . Read-write. Reset: 8h. HWM tuning parameter



**PM2x000000FE (FCH::PM2::test\_cntl)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000FE; PM2=FED8\_0400h

Bits	Description
7:4	Reserved.
3:0	<b>test_cntl</b> . Read-write. Reset: 0h. HWM tuning parameter

**PM2x000000FF (FCH::PM2::hwmmisccontro)**

Read-write. Reset: 00h.

\_aliasHOST; PM2x000000FF; PM2=FED8\_0400h

Bits	Description
7:6	Reserved.
5	<b>cycle_en</b> . Read-write. Reset: 0. HWM tuning parameter
4	<b>offset_can_en</b> . Read-write. Reset: 0. HWM tuning parameter
3	<b>gpio_a_cntrl</b> . Read-write. Reset: 0. HWM tuning parameter
2	Reserved.
1:0	<b>aport_mode</b> . Read-write. Reset: 0h. HWM tuning parameter

**7.3.10 GPIO Pin control registers****7.3.10.1 IOMUX Registers****7.3.10.1.1 IOMUX pin – EGPIO145/146/147/148 Programming Guideline**

The IOMUX register is used to select the function for multi-function IO pins. Pin information is listed in Table 102 [IOMUX Function Table].

When setting EGPIO145~148 pins for GPIO purpose, some extra settings are needed. Following are details.

- When EGPIO145 is used for GPIO purpose, please follow the following sequence
  - Set FCH::MISC::smitrappingrwrdoivr[i2c0sclmodeppod] = 0x1
  - Set FCH::MISC::smitrappingrwrdoivr[i2c0sdamodeppod] = 0x1
  - Set FCH::MISC::i2c0\_padctrl = 0x3C00\_0030
  - Set FCH::IOMUX::iomux145\_gpio = 0x2
- When EGPIO146 is used for GPIO purpose, please follow the following sequence.
  - Set FCH::MISC::smitrappingrwrdoivr[i2c0sclmodeppod] = 0x1
  - Set FCH::MISC::smitrappingrwrdoivr[i2c0sdamodeppod] = 0x1
  - Set FCH::MISC::i2c0\_padctrl = 0x3C00\_0030
  - Set FCH::IOMUX::iomux146\_gpio = 0x2
- When EGPIO147 is used for GPIO purpose, please follow the following sequence.:vsp
  - Set FCH::MISC::smitrappingrwrdoivr[i2c1sclmodeppod] = 0x1
  - Set FCH::MISC::smitrappingrwrdoivr[i2c1sdamodeppod] = 0x1
  - Set FCH::MISC::i2c1\_padctrl = 0x3C00\_0030
  - Set FCH::IOMUX::iomux147\_gpio = 0x2
- When EGPIO148 is used for GPIO purpose, please follow the following sequence.
  - Set FCH::MISC::smitrappingrwrdoivr[i2c1sclmodeppod] = 0x1
  - Set FCH::MISC::smitrappingrwrdoivr[i2c1sdamodeppod] = 0x1
  - Set FCH::MISC::i2c1\_padctrl = 0x3C00\_0030
  - Set FCH::IOMUX::iomux148\_gpio = 0x2

**7.3.10.1.2 IOMUX Funtional Table***Table 102: IOMUX Function Table*

IOMU X#	Bump Pin Name	Domain	GPIO #	GEVENT #	Override_0	Override_1	IOMU X == 0	IOMU X == 1	IOMU X == 2	IOMU X == 3	Default IO State	IOMU X value at reset
IOMU Xx0	BP_PWR_BTN_L/ AGPIO0	S5	0	21			PWR_BTN_L	GPIO0	GPIO0	GPIO0	PU	0
IOMU Xx1	BP_SYS_RESET_L/ AGPIO1	S5	1	19		RST_trap	SYS_RESET_L	GPIO1	GPIO1	GPIO1	PU	0
IOMU Xx2	BP_WAKE_L/ AGPIO2	S5	2	8			WAKE_L	GPIO2	GPIO2	GPIO2	PU	0
IOMU Xx3	BP_AGPIO3	S5	3	2			GPIO3	GPIO3	GPIO3	GPIO3	PU	0
IOMU Xx4	BP_AGPIO4	S5	4	4			GPIO4	GPIO4	GPIO4	GPIO4	PD	0
IOMU Xx5	BP_AGPIO5	S5	5	7			GPIO5	GPIO5	GPIO5	GPIO5	PD	0
IOMU Xx6	BP_AGPIO6	S5	6	10			GPIO6	GPIO6	GPIO6	GPIO6	PD	0
IOMU Xx7	BP_AGPIO7	S5	7	11			GPIO7	ZSC_tutter_rail	GPIO7	GPIO7	PU	0
IOMU Xx8	BP_AGPIO8 / TMU_CLK_OUT	S5	8	23			GPIO8	CIO_TMU_CLK_OUT0	CIO_TMU_CLK_OUT1	GPIO8	PD	0
IOMU Xx9	BP_AGPIO9	S5	9	22			GPIO9	GPIO9	GPIO9	GPIO9	PD	0
IOMU XxA	BP_S0A3_GPIO/ AGPIO10	S5	10				GPIO10	S0A3_GPIO	GPIO10	DF_VRCONTEXT_0	PU	0
IOMU XxB	BP_BLINK/ AGPI	S5	11	5			GPIO11	BLINK	GPIO11	GPIO11	PU	0

	O11											
IOMU XxC	BP_LL B_L/ AGPI O12	S5	12				LLB_ L	GPIO1 2	GPIO1 2	GPIO1 2	PU	0
IOMU Xx10	BP_U SB_O C0_L/ AGPI O16	S5	16	12			USB_ OC0_ L	GPIO1 6	GPIO1 6	GPIO1 6	PU	0
IOMU Xx11	BP_U SB_O C1_L/ AGPI O17	S5	17	13			USB_ OC1_ L	GPIO1 7	GPIO1 7	GPIO1 7	PU	0
IOMU Xx12	BP_U SB_O C2_L/ AGPI O18	S5	18	14			USB_ OC2_ L	GPIO1 8	GPIO1 8	GPIO1 8	PU	0
IOMU Xx13	BP_S MBUS 1_SCL / I2C3_ SCL/ I3C3_ SCL/ AGPI O19	S5	19				SMBU S1_SC L	I2C3_ SCL	I3C3_ SCL	GPIO1 9	n/a	0
IOMU Xx14	BP_S MBUS 1_SD A/ I2C3_ SDA/ I3C3_ SDA/ AGPI O20	S5	20				SMBU S1_SD A	I2C3_ SDA	I3C3_ SDA	GPIO2 0	n/a	0
IOMU Xx15	BP_ES PI_RE SET_L / KBRS T_L/ AGPI O21	S5	21				ESPI_ RESE T_L	KBRS T_L	GPIO2 1	GPIO2 1	PU	0
IOMU Xx16	BP_ES PI_AL ERT_L _ESPI	S5	22	3		ESPI_ ALER T_L	ESPI_ ALER T_D1	GPIO2 2	GPIO2 2	SD0_C MD	PU	0

	_IO1/ AGPI O22											
IOMU Xx17	BP_A C_PR ES/ AGPI O23	S5	23	16			AC_P RES	GPIO2 3	GPIO2 3	GPIO2 3	PU	0
IOMU Xx18	BP_U SB_O C3_L/ AGPI O24	S5	24	15			USB_ OC3_ L	GPIO2 4	GPIO2 4	GPIO2 4	PU	0
IOMU Xx1A	BP_PC IE_RS T0_L/ EGPIO 26	S5	26				PCIE_ RST0_ L	GPIO2 6	GPIO2 6	GPIO2 6	n/a	0
IOMU Xx1B	BP_A GPIO2 7/ PCIE_ RST1_ L	S5	27				GPIO2 7	PCIE_ RST1_ L	GPIO2 7	GPIO2 7	PD	0
IOMU Xx1D	BP_SP I_TPM _CS_L / AGPI O29	S5	29	9			SPI_T PM_C S_L	GPIO2 9	GPIO2 9	GPIO2 9	PU	0
IOMU Xx1E	BP_SP I_CS2 _L/ ESPI_ CS_L/ AGPI O30	S5	30				SPI_C S2_L	ESPI_ CS_L	GPIO3 0	GPIO3 0	PU	1
IOMU Xx1F	BP_SP I_CS3 _L/ AGPI O31/ BP_SP I2_CS 3	S5	31				SPI_C S3_L	GPIO3 1	GPIO3 1	SPI2_ CS3_L	PU	0
IOMU Xx20	BP_A GPIO3 2	S5	32	17			GPIO3 2	LPC_ RST_L	GPIO3 2	GPIO3 2	n/a	1
IOMU Xx26	BP_C LK_R EQ5_L /	S5	38				CLK_ REQ5_ L	GPIO3 8	GPIO3 8	GPIO3 8	PU	0

	AGPI O38											
IOMU Xx27	BP_C LK_R EQ6_L / AGPI O39	S5	39				CLK_ REQ6 _L	GPIO3 9	GPIO3 9	GPIO3 9	PU	0
IOMU Xx28	BP_A GPIO4 0	S5	40	20			GPIO4 0	GPIO4 0	GPIO4 0	GPIO4 0	PD	0
IOMU Xx2A	BP_A GPIO4 2	S5	42				GPIO4 2	DF_V RCON TEXT _1	GPIO4 2	GPIO4 2	PU	0
IOMU Xx43	BP_SP I_RO M_RE Q/ EGPIO 67	S5	67				SPI_R OM_R EQ	GPIO6 7	GPIO6 7	GPIO6 7	PD	0
IOMU Xx44	BP_SP I1_DA T[2]/ ESPI_ DAT[2 ]/ AGPI O68	S0	68				SPI1_ DAT2	GPIO6 8		SD0_ DATA 3	PU	0
IOMU Xx45	BP_SP I1_DA T[3]/ ESPI_ DAT[3 ]/ AGPI O69	S0	69				SPI1_ DAT3	GPIO6 9	SD0_C LK	GPIO6 9	PU	0
IOMU Xx46	BP_SP I2_CL K/ EGPIO 70	S0	70			ROMT YPE_s trap	SPI2_ CLK	GPIO7 0	GPIO7 0	GPIO7 0	PD	0
IOMU Xx4A	BP_SP I1_CS 1_L/ GFX1 0_CA C_IPI O0/ EGPIO 74	S0	74				SPI1_ CS1_L	GPIO7 4	GFX1 0_CA C_IPI O0	GPIO7 4	PU	0
IOMU	BP_SP	S0	75				SPI2_	GPIO7	GPIO7	GPIO7	PU	0

Xx4B	I2_CS 1_L/ EGPIO 75						CS1_L	5	5	5		
IOMU Xx4C	BP_SP I_RO M_GN T/ EGPIO 76	S5	76				SPI_R OM_G NT	GPIO7 6	GPIO7 6	GPIO7 6	PD	0
IOMU Xx4D	BP_SP I1_CL K/ ESPI_ CLK/ EGPIO 77	S0	77			ROMT YPE1_ strap	SPI1_ CLK	GPIO7 7	GPIO7 7	SD0_ DATA 0	PD	0
IOMU Xx4E	BP_SP I1_CS 2_L/ GFX1 0_CA C_IPI O1/ EGPIO 78	S0	78				SPI1_ CS2_L	GPIO7 8	GFX1 0_CA C_IPI O1	SD0_ DATA 1	PU	0
IOMU Xx4F	BP_SP I1_CS 3_L/ EGPIO 79/ BP_SP I2_CS 2_L	S0	79				SPI1_ CS3_L	GPIO7 9	SPI2_ CS2_L	GPIO7 9	PU	0
IOMU Xx50	BP_SP I1_DA T[1]/ ESPI_ DAT[1 ]/ EGPIO 80	S0	80				SPI1_ DAT1	GPIO8 0	GPIO8 0	SD0_ DATA 2	PU	0
IOMU Xx51	BP_SP I1_DA T[0]/ ESPI_ DAT[0 ]/ EGPIO 81	S0	81				SPI1_ DAT0	GPIO8 1	GPIO8 1	GPIO8 1	PU	0
IOMU Xx54	BP_FA NIN0/	S0	84	18			FANI N0	GPIO8 4	GPIO8 4	GPIO8 4	PU	0

	AGPI O84											
IOMU Xx55	BP_FA NOUT 0/ AGPI O85	S0	85			FanOu t_intB	FANO UT0	GPIO8 5	GPIO8 5	GPIO8 5	PU	1
IOMU Xx56	BP_P KGID 2	S0	86			PKG_s trap2	GPIO8 6	GPIO8 6	GPIO8 6	GPIO8 6	PU	0
IOMU Xx59	BP_G ENIN T1_L/ PSP_I NTR0/ AGPI O89	S0	89	0			GENI NT1_L	PSP_I NTR0	GPIO8 9	GPIO8 9	PU	0
IOMU Xx5A	BP_G ENIN T2_L/ PSP_I NTR1/ AGPI O90	S0	90	1			GENI NT2_L	PSP_I NTR1	GPIO9 0	GPIO9 0	PU	0
IOMU Xx5B	BP_SP KR/ AGPI O91	S0	91	6			SPKR	GPIO9 1	GPIO9 1	GPIO9 1	PD	1
IOMU Xx5C	BP_C LK_R EQ0_L	S0	92				CLK_ REQ0 _L	GPIO9 2	GPIO9 2	GPIO9 2	PU	0
IOMU Xx68	BP_SP I2_DA T[0]/ EGPIO 104	S0	104				SPI2_ DAT0	GPIO1 04	GPIO1 04	GPIO1 04	PD	0
IOMU Xx69	BP_SP I2_DA T[1]/ EGPIO 105	S0	105				SPI2_ DAT1	GPIO1 05	GPIO1 05	GPIO1 05	PD	0
IOMU Xx6A	BP_SP I2_DA T[2]/ EGPIO 106	S0	106				SPI2_ DAT2	GPIO1 06	GPIO1 06	GPIO1 06	PU	0
IOMU Xx6B	BP_SP I2_DA T[3]/ EGPIO 107	S0	107				SPI2_ DAT3	GPIO1 07	GPIO1 07	GPIO1 07	PU	0
IOMU	BP_S	S0	113				SMBU	I2C2_ I3C2_ GPIO1			n/a	0

Xx71	MBUS 0_SCL / I2C2_ SCL/ I3C2_ SCL/ EGPIO 113						S0_SC L	SCL	SCL	13		
IOMU Xx72	BP_S MBUS 0_SD A/ I2C2_ SDA/ I3C2_ SDA/ EGPIO 114	S0	114				SMBU S0_SD A	I2C2_ SDA	I3C2_ SDA	GPIO1 14	n/a	0
IOMU Xx73	BP_C LK_R EQ1_L / AGPI O115	S0	115				CLK_ REQ1 _L	GPIO1 15	GPIO1 15	GPIO1 15	PU	0
IOMU Xx74	BP_C LK_R EQ2_L / AGPI O116	S0	116				CLK_ REQ2 _L	GPIO1 16	GPIO1 16	GPIO1 16	PU	0
IOMU Xx82	BP_A GPIO1 30	S0	130				GPIO1 30	GPIO1 30	GPIO1 30	GPIO1 30	PU	0
IOMU Xx83	BP_C LK_R EQ3_L	S0	131				CLK_ REQ3 _L	GPIO1 31	GPIO1 31	GPIO1 31	PU	0
IOMU Xx84	BP_C LK_R EQ4_L / OSCI N/ EGPIO 132	S0	132				CLK_ REQ4 _L	OSCI N	GPIO1 32	GPIO1 32	PU	0
IOMU Xx87	BP_E GPIO1 35/ UART 2_CTS _L/ UART	S0	135				GPIO1 35	UART 2_CTS _L	UART 3_TX D	GPIO1 35	PD	0



	3_TX D											
IOMU Xx88	BP_E GPIO1 36/ UART 2_RX D	S0	136				GPIO1 36	UART 2_RX D	GPIO1 36	GPIO1 36	PD	0
IOMU Xx89	BP_E GPIO1 37/ UART 2_RTS _L/ UART 3_RX D	S0	137				GPIO1 37	UART 2_RTS _L	UART 3_RX D	GPIO1 37	PU	0
IOMU Xx8A	BP_E GPIO1 38/ UART 2_TX D	S0	138				GPIO1 38	UART 2_TX D	GPIO1 38	GPIO1 38	PU	0
IOMU Xx8B	BP_A GPIO1 39/ UART 2_INT R	S0	139				GPIO1 39	UART 2_INT R	GPIO1 39	GPIO1 39	PD	0
IOMU Xx8C	BP_E GPIO1 40/ UART 0_CTS _L/ UART 1_TX D	S0	140				GPIO1 40	UART 0_CTS _L	UART 1_TX D	GPIO1 40	PD	0
IOMU Xx8D	BP_E GPIO1 41/ UART 0_RX D	S0	141				GPIO1 41	UART 0_RX D	GPIO1 41	GPIO1 41	PD	0
IOMU Xx8E	BP_E GPIO1 42/ UART 0_RTS _L/ UART 1_RX	S0	142				GPIO1 42	UART 0_RTS _L	UART 1_RX D	GPIO1 42	PU	0

	D											
IOMU Xx8F	BP_E GPIO1 43/ UART 0_TX D	S0	143				GPIO1 43	UART 0_TX D	GPIO1 43	GPIO1 43	PU	0
IOMU Xx90	BP_A GPIO1 44/ SHUT DOW N_L/ UART 0_INT R	S0	144				GPIO1 44	ShutD own_L	UART 0_INT R	GPIO1 44	PD	0
IOMU Xx91	BP_I2 C0_SC L/ I3C0_ SCL/ EGPIO 145	S0	145				I2C0_ SCL	I3C0_ SCL	GPIO1 45	GPIO1 45	n/a	0
IOMU Xx92	BP_I2 C0_SD A/ I3C0_ SDA/ EGPIO 146	S0	146				I2C0_ SDA	I3C0_ SDA	GPIO1 46	GPIO1 46	n/a	0
IOMU Xx93	BP_I2 C1_SC L/ I3C1_ SCL/ EGPIO 147	S0	147				I2C1_ SCL	I3C1_ SCL	GPIO1 47	GPIO1 47	n/a	0
IOMU Xx94	BP_I2 C1_SD A/ I3C1_ SDA/ EGPIO 148	S0	148				I2C1_ SDA	I3C1_ SDA	GPIO1 48	GPIO1 48	n/a	0
IOMU Xx99	BP_E GPIO1 53/ UART 4_CTS _L	S0	153				GPIO1 53	UART 4_CTS _L	GPIO1 53	GPIO1 53	PD	0
IOMU Xx9A	BP_E GPIO1	S0	154				GPIO1 54	UART 4_RTS	GPIO1 54	GPIO1 54	PU	0

	54/ UART 4_RTS _L							_L				
IOMU Xx9B	BP_E GPIO1 55/ UART 4_RX D	S0	155				GPIO1 55	UART 4_RX D	GPIO1 55	GPIO1 55	PD	0
IOMU Xx9C	BP_E GPIO1 56/ UART 4_TX D	S0	156				GPIO1 56	UART 4_TX D	GPIO1 56	GPIO1 56	PU	0
IOMU Xx9D	BP_A GPIO1 57/ UART 4_INT R	S0	157				GPIO1 57	UART 4_INT R	GPIO1 57	GPIO1 57	PD	0

IOMUX registers are accessed by memory-mapped (or IO-mapped) Ios. And they range from "AcpiMMioAddr" + 0xD00 to "AcpiMMioAddr" + 0xDFF. AcpiMMioAddr is FED8\_0000 or the address defined in PMxD6.

The IOMUX register is used to select the function for multi-function IO pins.

Note:

1. PWR\_BTN\_L\_AGPI00 can only be used as PWR\_BTN\_. Since ACPI5.0 require Power Button be claimed as GPIO, and BIOS need the GPIO number to program its debouncing time, thus AGPI00 is assigned.
2. when UART Input is selected (UART\_CTS\_L, UART\_RXD, UART\_RTS\_N, UART\_INTR), corresponding GPIOOutEn bit need be set to '0'
3. LPCCLK0 (LPCCLK0/EMMC\_DATA4/EGPIO74) is used to supply clock to internal LPC logic, if LPCCLK0 is disabled (IOMUX-74 not equal 00b, or LPC PCI\_CFG\_regxD0[13]=0), any transaction to LPC bus will cause system to lock up. Sending transaction to SPI/eSPI is OK.
4. Pad PCIE\_RST\_L\_EGPIO26 is driven by (oPcieRstBToPad\_S5 & S5GpioOut[26]) and pad EGPIO27\_PCIE\_RST1\_L is driven by (oPcieRstBToPad\_S5 & S5GpioOut[27]), this is to provide software to have control on deassertion of PCIE\_RST\_L. Since S5GpioOut[26] and S5GpioOut[27] will be reset to '0' by PciRstB, software need to set S5GpioOut[26]=1 and S5GpioOut[27]=1 to deassert PCIE\_RST\_L and PCIE\_RST1\_L.

**IOMUXx00000000 (FCH::IOMUX::iomux0\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000000; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000001 (FCH::IOMUX::iomux1\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000001; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000002 (FCH::IOMUX::iomux2\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000002; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000003 (FCH::IOMUX::iomux3\_gpio)**

Read-write. Reset: 00h.

**\*Note:**

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000003; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000004 (FCH::IOMUX::iomux4\_gpio)**

Read-write. Reset: 00h.

**\*Note:**

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000004; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000005 (FCH::IOMUX::iomux5\_gpio)**

Read-write. Reset: 00h.

**\*Note:**

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000005; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000006 (FCH::IOMUX::iomux6\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000006; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000007 (FCH::IOMUX::iomux7\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000007; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000008 (FCH::IOMUX::iomux8\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000008; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000009 (FCH::IOMUX::iomux9\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000009; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000000A (FCH::IOMUX::iomux10\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000000A; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000000B (FCH::IOMUX::iomux11\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000000B; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000000C (FCH::IOMUX::iomux12\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000000C; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000000D (FCH::IOMUX::iomux13\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000000D; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000000E (FCH::IOMUX::iomux14\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000000E; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3



**IOMUXx0000000F (FCH::IOMUX::iomux15\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000000F; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000010 (FCH::IOMUX::iomux16\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000010; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000011 (FCH::IOMUX::iomux17\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000011; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000012 (FCH::IOMUX::iomux18\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000012; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000013 (FCH::IOMUX::iomux19\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000013; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000014 (FCH::IOMUX::iomux20\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000014; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000015 (FCH::IOMUX::iomux21\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000015; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000016 (FCH::IOMUX::iomux22\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000016; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000017 (FCH::IOMUX::iomux23\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000017; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000018 (FCH::IOMUX::iomux24\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000018; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000019 (FCH::IOMUX::iomux25\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000019; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000001A (FCH::IOMUX::iomux26\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000001A; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000001B (FCH::IOMUX::iomux27\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000001B; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000001C (FCH::IOMUX::iomux28\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000001C; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000001D (FCH::IOMUX::iomux29\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000001D; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000001E (FCH::IOMUX::iomux30\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000001E; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000001F (FCH::IOMUX::iomux31\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000001F; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000020 (FCH::IOMUX::iomux32\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000020; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000021 (FCH::IOMUX::iomux33\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000021; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000022 (FCH::IOMUX::iomux34\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000022; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000023 (FCH::IOMUX::iomux35\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000023; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000024 (FCH::IOMUX::iomux36\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000024; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000025 (FCH::IOMUX::iomux37\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000025; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000026 (FCH::IOMUX::iomux38\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000026; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3



**IOMUXx00000027 (FCH::IOMUX::iomux39\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000027; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000028 (FCH::IOMUX::iomux40\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000028; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000029 (FCH::IOMUX::iomux41\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000029; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000002A (FCH::IOMUX::iomux42\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000002A; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000002B (FCH::IOMUX::iomux43\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000002B; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000002C (FCH::IOMUX::iomux44\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000002C; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000002D (FCH::IOMUX::iomux45\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000002D; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000002E (FCH::IOMUX::iomux46\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000002E; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000002F (FCH::IOMUX::iomux47\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000002F; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000030 (FCH::IOMUX::iomux48\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000030; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000031 (FCH::IOMUX::iomux49\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000031; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000032 (FCH::IOMUX::iomux50\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000032; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000033 (FCH::IOMUX::iomux51\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000033; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000034 (FCH::IOMUX::iomux52\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000034; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000035 (FCH::IOMUX::iomux53\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000035; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000036 (FCH::IOMUX::iomux54\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000036; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000037 (FCH::IOMUX::iomux55\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000037; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000038 (FCH::IOMUX::iomux56\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000038; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000039 (FCH::IOMUX::iomux57\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000039; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000003A (FCH::IOMUX::iomux58\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000003A; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000003B (FCH::IOMUX::iomux59\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000003B; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000003C (FCH::IOMUX::iomux60\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000003C; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000003D (FCH::IOMUX::iomux61\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000003D; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000003E (FCH::IOMUX::iomux62\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000003E; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3



**IOMUXx0000003F (FCH::IOMUX::iomux63\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000003F; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000040 (FCH::IOMUX::iomux64\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000040; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000041 (FCH::IOMUX::iomux65\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000041; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000042 (FCH::IOMUX::iomux66\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000042; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000043 (FCH::IOMUX::iomux67\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000043; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000044 (FCH::IOMUX::iomux68\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000044; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000045 (FCH::IOMUX::iomux69\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000045; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000046 (FCH::IOMUX::iomux70\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000046; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000047 (FCH::IOMUX::iomux71\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000047; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000048 (FCH::IOMUX::iomux72\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000048; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000049 (FCH::IOMUX::iomux73\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000049; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000004A (FCH::IOMUX::iomux74\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000004A; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000004B (FCH::IOMUX::iomux75\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000004B; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000004C (FCH::IOMUX::iomux76\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000004C; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000004D (FCH::IOMUX::iomux77\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000004D; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000004E (FCH::IOMUX::iomux78\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000004E; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000004F (FCH::IOMUX::iomux79\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000004F; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000050 (FCH::IOMUX::iomux80\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000050; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000051 (FCH::IOMUX::iomux81\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000051; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000052 (FCH::IOMUX::iomux82\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000052; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000053 (FCH::IOMUX::iomux83\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000053; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000054 (FCH::IOMUX::iomux84\_gpio)**

Read-write. Reset: 00h.

**\*Note:**

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000054; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000055 (FCH::IOMUX::iomux85\_gpio)**

Read-write. Reset: 00h.

**\*Note:**

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000055; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000056 (FCH::IOMUX::iomux86\_gpio)**

Read-write. Reset: 00h.

**\*Note:**

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000056; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3



**IOMUXx00000057 (FCH::IOMUX::iomux87\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000057; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000058 (FCH::IOMUX::iomux88\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000058; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000059 (FCH::IOMUX::iomux89\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000059; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000005A (FCH::IOMUX::iomux90\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000005A; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000005B (FCH::IOMUX::iomux91\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000005B; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000005C (FCH::IOMUX::iomux92\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000005C; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000005D (FCH::IOMUX::iomux93\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000005D; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000005E (FCH::IOMUX::iomux94\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000005E; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000005F (FCH::IOMUX::iomux95\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000005F; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000060 (FCH::IOMUX::iomux96\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000060; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000061 (FCH::IOMUX::iomux97\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000061; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000062 (FCH::IOMUX::iomux98\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000062; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000063 (FCH::IOMUX::iomux99\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000063; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000064 (FCH::IOMUX::iomux100\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000064; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000065 (FCH::IOMUX::iomux101\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000065; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000066 (FCH::IOMUX::iomux102\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000066; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000067 (FCH::IOMUX::iomux103\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000067; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000068 (FCH::IOMUX::iomux104\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000068; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000069 (FCH::IOMUX::iomux105\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000069; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000006A (FCH::IOMUX::iomux106\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000006A; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000006B (FCH::IOMUX::iomux107\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000006B; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000006C (FCH::IOMUX::iomux108\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000006C; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000006D (FCH::IOMUX::iomux109\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000006D; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000006E (FCH::IOMUX::iomux110\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000006E; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3



**IOMUXx0000006F (FCH::IOMUX::iomux111\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000006F; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000070 (FCH::IOMUX::iomux112\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000070; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000071 (FCH::IOMUX::iomux113\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000071; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000072 (FCH::IOMUX::iomux114\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000072; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000073 (FCH::IOMUX::iomux115\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000073; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000074 (FCH::IOMUX::iomux116\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000074; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000075 (FCH::IOMUX::iomux117\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000075; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000076 (FCH::IOMUX::iomux118\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000076; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000077 (FCH::IOMUX::iomux119\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000077; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000078 (FCH::IOMUX::iomux120\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000078; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000079 (FCH::IOMUX::iomux121\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000079; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000007A (FCH::IOMUX::iomux122\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000007A; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000007B (FCH::IOMUX::iomux123\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000007B; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000007C (FCH::IOMUX::iomux124\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000007C; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000007D (FCH::IOMUX::iomux125\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000007D; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000007E (FCH::IOMUX::iomux126\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000007E; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000007F (FCH::IOMUX::iomux127\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000007F; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000080 (FCH::IOMUX::iomux128\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000080; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000081 (FCH::IOMUX::iomux129\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000081; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000082 (FCH::IOMUX::iomux130\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000082; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000083 (FCH::IOMUX::iomux131\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000083; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000084 (FCH::IOMUX::iomux132\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000084; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000085 (FCH::IOMUX::iomux133\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000085; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000086 (FCH::IOMUX::iomux134\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000086; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3



**IOMUXx00000087 (FCH::IOMUX::iomux135\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000087; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000088 (FCH::IOMUX::iomux136\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000088; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000089 (FCH::IOMUX::iomux137\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000089; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000008A (FCH::IOMUX::iomux138\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000008A; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000008B (FCH::IOMUX::iomux139\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000008B; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000008C (FCH::IOMUX::iomux140\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000008C; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000008D (FCH::IOMUX::iomux141\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000008D; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000008E (FCH::IOMUX::iomux142\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000008E; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000008F (FCH::IOMUX::iomux143\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000008F; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000090 (FCH::IOMUX::iomux144\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000090; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000091 (FCH::IOMUX::iomux145\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000091; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000092 (FCH::IOMUX::iomux146\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000092; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000093 (FCH::IOMUX::iomux147\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000093; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000094 (FCH::IOMUX::iomux148\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000094; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000095 (FCH::IOMUX::iomux149\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000095; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000096 (FCH::IOMUX::iomux150\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000096; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000097 (FCH::IOMUX::iomux151\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000097; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000098 (FCH::IOMUX::iomux152\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000098; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx00000099 (FCH::IOMUX::iomux153\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx00000099; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000009A (FCH::IOMUX::iomux154\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000009A; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000009B (FCH::IOMUX::iomux155\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000009B; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000009C (FCH::IOMUX::iomux156\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000009C; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**IOMUXx0000009D (FCH::IOMUX::iomux157\_gpio)**

Read-write. Reset: 00h.

\*Note:

&lt;N&gt; denotes number in hexadecimal: 00h ~ 9Dh.

&lt;X&gt; denotes number in decimal: 0 ~ 157.

\_aliasHOST; IOMUXx0000009D; IOMUX=FED8\_0D00h

Bits	Description
7:2	Reserved.
1:0	<b>iomux_gpio_x.</b> Read-write. Reset: 0h. <b>Description:</b> Multi-function IO pin function select for pin GPIO<X> 00: function-0 01: function-1 10: function-2 11: function-3

**7.3.10.2 GPIO Registers**

The GPIO pins are controlled by a combination of device enables and by their specific GPIO and IOMUX register pair.

GPIO registers are accessed through memory-mapped ACPIMMIO region. The offset is relative to FED8\_0000h+1500h. GPIO bank 0 registers range from FED8\_0000h+1500h to FED8\_0000h+15FFh. GPIO Bank 1 registers range from FED8\_0000h+1600h to FED8\_0000h+16FFh. GPIO Bank 2 registers range from FED8\_0000h+1700h to FED8\_0000h+17FFh. GPIO Bank 3 registers range from FED8\_0000h+1800h to FED8\_0000h+18FFh.

*Table 103: I2C Pad Configuration Method*

	PAD name	GPIO register (bank, offset) (No use)	Misc_Reg (Control I2C PAD)
I2C0	I2C0_SCL_EGPIO145_I3C0_SCL	0x244	0xD8
	I2C0_SDA_EGPIO146_I3C0_SDA	0x248	
I2C1	I2C1_SCL_EGPIO147_I3C1_SCL	0x24C	0xDC
	I2C1_SDA_EGPIO148_I3C1_SDA	0x250	
I2C2	SMBUS0_SCL_EGPIO113_I2C2_SCL_I3C2_SCL	0x1C4	0xE0
	SMBUS0_SDA_EGPIO114_I2C2_SDA_I3C2_SD	0x1C8	



	A		
I2C3	SMBUS1_SCL_AGPIO19_I2C3_SCL_I3C3_SCL	0x04C	0xE4
	SMBUS1_SDA_AGPIO20_I2C3_SDA_I3C3_SDA	0x050	

Table 104: GPIO BANK0 Register Default Value

Register	Reset Value	Index	Function
GPIOx000	0014_0000h	0	BP_PWR_BTN_L_AGPIO0
GPIOx004	0014_0000h	1	BP_RST_strap_SYS_RESET_L_AGPIO1
GPIOx008	0014_0000h	2	BP_WAKE_L_AGPIO2
GPIOx00C	0014_0000h	3	BP_AGPIO3
GPIOx010	0024_0000h	4	BP_AGPIO4
GPIOx014	0024_0000h	5	BP_AGPIO5
GPIOx018	0024_0000h	6	BP_AGPIO6
GPIOx01C	0014_0000h	7	BP_AGPIO7
GPIOx020	0024_0000h	8	BP_AGPIO8_TMU_CLK_OUT0_TMU_CLK_OUT1
GPIOx024	0024_0000h	9	BP_AGPIO9
GPIOx028	0014_0000h	10	BP_AGPIO10_S0A3_GPIO
GPIOx02C	0014_0000h	11	BP_AGPIO11_BLINK
GPIOx030	0014_0000h	12	BP_LLБ_L_AGPIO12
GPIOx034	0000_0000h	13	Reserved
GPIOx038	0000_0000h	14	Reserved
GPIOx03C	0000_0000h	15	Reserved
GPIOx040	0014_0000h	16	BP_USB_OC0_L_AGPIO16
GPIOx044	0014_0000h	17	BP_USB_OC1_L_AGPIO17
GPIOx048	0014_0000h	18	BP_USB_OC2_L_AGPIO18
GPIOx04C	0000_0000h	19	BP_SMBUS1_SCL_AGPIO19_I2C3_SCL_I3C3_SCL
GPIOx050	0006_0000h	20	BP_SMBUS1_SDA_AGPIO20_I2C3_SDA_I3C3_SDA
GPIOx054	0014_0000h	21	BP_ESPI_RESET_L_AGPIO21_KBRST_L
GPIOx058	0024_0000h	22	BP_ESPI_ALERT_L_ESPI_IO1_AGPIO22_ESPI_ALE_RT_D1_SD0_CMD
GPIOx05C	0014_0000h	23	BP_AC_PRES_AGPIO23
GPIOx060	0014_0000h	24	BP_USB_OC3_L_AGPIO24
GPIOx064	0000_0000h	25	Reserved
GPIOx068	0024_0000h	26	BP_PCIE_RST0_L_AGPIO26
GPIOx06C	0024_0000h	27	BP_AGPIO27_PCIE_RST1_L
GPIOx070	0000_0000h	28	Reserved
GPIOx074	0014_0000h	29	BP_SPI_TPM_CS_L_AGPIO29
GPIOx078	0014_0000h	30	BP_SPI_CS2_L_AGPIO30_ESPI_CS_L
GPIOx07C	0014_0000h	31	BP_SPI_CS3_L_AGPIO31
GPIOx080	0024_0000h	32	BP_AGPIO32_LPC_RST_L
GPIOx084	0000_0000h	33	Reserved
GPIOx088	0000_0000h	34	Reserved
GPIOx08C	0000_0000h	35	Reserved
GPIOx090	0000_0000h	36	Reserved
GPIOx094	0000_0000h	37	Reserved
GPIOx098	0014_0000h	38	BP_CLK_REQ5_L_AGPIO38
GPIOx09C	0014_0000h	39	BP_CLK_REQ6_L_AGPIO39
GPIOx0A0	0024_0000h	40	BP_AGPIO40

GPIOx0A4	0000_0000h	41	Reserved
GPIOx0A8	0014_0000h	42	BP_AGPIO42
GPIOx0AC	0000_0000h	43	Int_FakeSts0
GPIOx0B0	0000_0000h	44	Int_ShdwSysAlarmFire
GPIOx0B4	0000_0000h	45	Int_pwr_bttn (NOTE: This Event need to be debounced)
GPIOx0B8	0000_0000h	46	Int_FakeSts0_2nd
GPIOx0BC	0000_0000h	47	Int_Wake_up_WDT0
GPIOx0C0	0000_0000h	48	Int_DhcpLeaseTimerExpire
GPIOx0C4	0000_0000h	49	Int_ASFSlaveIntr
GPIOx0C8	0000_0000h	50	Int_sm_irq_
GPIOx0CC	0000_0000h	51	Int_WakeFromLLB
GPIOx0D0	0000_0000h	52	Int_AcDcTimerEvent
GPIOx0D4	0000_0000h	53	Int_ALTHPET_TimerSts
GPIOx0D8	0000_0000h	54	Int_iSocEvent0
GPIOx0DC	0000_0000h	55	Int_iSocEvent1
GPIOx0E0	0000_0000h	56	Int_iSocEvent2
GPIOx0E4	0000_0000h	57	Int_iSocEvent3
GPIOx0E8	0000_0000h	58	Int_usb_xhc_0_acpi_pme
GPIOx0EC	0000_0000h	59	Int_usb_xhc_1_acpi_pme
GPIOx0F0	0000_0000h	60	Int_RTC_STS_reg
GPIOx0F4	0000_0000h	61	Int_ACP_FCH_AZ_Wake
GPIOx0F8	0000_0000h	62	Int_ACP_FCH_I2S_Wake
GPIOx0FC	ff00_0000h	63	GPIO Wake/Interrupt master switch

Table 105: GPIO BANK1 Register Default Value

Register	Reset Value	Index	Function
GPIOx100	0000_0000h	64	Reserved
GPIOx104	0000_0000h	65	Reserved
GPIOx108	0000_0000h	66	Reserved
GPIOx10C	0024_0000h	67	BP_SPI_ROM_REQ_EGPIO67
GPIOx110	0014_0000h	68	BP_SPI1_DAT2_AGPIO68_SD0_DATA3
GPIOx114	0014_0000h	69	BP_SPI1_DAT3_AGPIO69_SD0_CLK
GPIOx118	0024_0000h	70	BP_SPI2_CLK_EGPIO70
GPIOx11C	0000_0000h	71	Reserved
GPIOx120	0000_0000h	72	Reserved
GPIOx124	0000_0000h	73	Reserved
GPIOx128	0014_0000h	74	BP_SPI1_CS1_L_EGPIO74_GFX10_CAC_IPIO0
GPIOx12C	0014_0000h	75	BP_SPI2_CS1_L_EGPIO75
GPIOx130	0024_0000h	76	BP_SPI_ROM_GNT_EGPIO76
GPIOx134	0024_0000h	77	BP_SPI1_CLK_EGPIO77_SD0_DATA0
GPIOx138	0014_0000h	78	BP_SPI1_CS2_L_EGPIO78_GFX10_CAC_IPIO1_SD0_DATA1
GPIOx13C	0014_0000h	79	BP_SPI1_CS3_L_EGPIO79
GPIOx140	0024_0000h	80	BP_SPI1_DAT1_EGPIO80_SD0_DATA2
GPIOx144	0024_0000h	81	BP_SPI1_DAT0_EGPIO81
GPIOx148	0000_0000h	82	Reserved
GPIOx14C	0000_0000h	83	Reserved
GPIOx150	0014_0000h	84	BP_FANIN0_AGPIO84

GPIOx154	0014_0000h	85	BP_FANOUT0_AGPIO85
GPIOx158	0014_0000h	86	BP_PKG_strap2_AGPIO86
GPIOx15C	0000_0000h	87	Reserved
GPIOx160	0000_0000h	88	Reserved
GPIOx164	0014_0000h	89	BP_GENINT1_L_AGPIO89_PSP_INTR0
GPIOx168	0014_0000h	90	BP_GENINT2_L_AGPIO90_PSP_INTR1
GPIOx16C	0024_0000h	91	BP_SPKR_AGPIO91
GPIOx170	0014_0000h	92	BP_CLK_REQ0_L_AGPIO92
GPIOx174	0000_0000h	93	Reserved
GPIOx178	0000_0000h	94	Reserved
GPIOx17C	0000_0000h	95	Reserved
GPIOx180	0000_0000h	96	Reserved
GPIOx184	0000_0000h	97	Reserved
GPIOx188	0000_0000h	98	Reserved
GPIOx18C	0000_0000h	99	Reserved
GPIOx190	0000_0000h	100	Reserved
GPIOx194	0000_0000h	101	Reserved
GPIOx198	0000_0000h	102	Reserved
GPIOx19C	0000_0000h	103	Reserved
GPIOx1A0	0024_0000h	104	BP_SPI2_DAT0_EGPIO104
GPIOx1A4	0024_0000h	105	BP_SPI2_DAT1_EGPIO105
GPIOx1A8	0014_0000h	106	BP_SPI2_DAT2_EGPIO106
GPIOx1AC	0014_0000h	107	BP_SPI2_DAT3_EGPIO107
GPIOx1B0	0000_0000h	108	Reserved
GPIOx1B4	0000_0000h	109	Reserved
GPIOx1B8	0000_0000h	110	Reserved
GPIOx1BC	0000_0000h	111	Reserved
GPIOx1C0	0000_0000h	112	Reserved
GPIOx1C4	0010_0000h	113	BP_SMBUS0_SCL_EGPIO113_I2C2_SCL_I3C2_SCL
GPIOx1C8	0016_0000h	114	BP_SMBUS0_SDA_EGPIO114_I2C2_SDA_I3C2_SDA
GPIOx1CC	0014_0000h	115	BP_CLK_REQ1_L_AGPIO115
GPIOx1D0	0014_0000h	116	BP_CLK_REQ2_L_AGPIO116
GPIOx1D4	0000_0000h	117	Reserved
GPIOx1D8	0000_0000h	118	Reserved
GPIOx1DC	0000_0000h	119	Reserved
GPIOx1E0	0000_0000h	120	Reserved
GPIOx1E4	0000_0000h	121	Reserved
GPIOx1E8	0000_0000h	122	Reserved
GPIOx1EC	0000_0000h	123	Reserved
GPIOx1F0	0000_0000h	124	Reserved
GPIOx1F4	0000_0000h	125	Reserved
GPIOx1F8	0000_0000h	126	Reserved
GPIOx1FC	0000_0000h	127	Reserved

Table 106: GPIO BANK2 Register Default Value

Register	Reset Value	Index	Function
GPIOx200	0000_0000h	128	Reserved
GPIOx204	0000_0000h	129	Reserved

GPIOx208	0014_0000h	130	BP_AGPIO130
GPIOx20C	0014_0000h	131	BP_CLK_REQ3_L_EGPIO131
GPIOx210	0014_0000h	132	BP_CLK_REQ4_L_EGPIO132_OSCIN
GPIOx214	0000_0000h	133	Reserved
GPIOx218	0000_0000h	134	Reserved
GPIOx21C	0024_0000h	135	BP_EGPIO135_UART2_CTS_L_UART3_TXD
GPIOx220	0024_0000h	136	BP_EGPIO136_UART2_RXD
GPIOx224	0014_0000h	137	BP_EGPIO137_UART2_RTS_L_UART3_RXD
GPIOx228	0014_0000h	138	BP_EGPIO138_UART2_TXD
GPIOx22C	0024_0000h	139	BP_AGPIO139_UART2_INTR
GPIOx230	0024_0000h	140	BP_EGPIO140_UART0_CTS_L_UART1_TXD
GPIOx234	0024_0000h	141	BP_EGPIO141_UART0_RXD
GPIOx238	0014_0000h	142	BP_EGPIO142_UART0_RTS_L_UART1_RXD
GPIOx23C	0014_0000h	143	BP_EGPIO143_UART0_TXD
GPIOx240	0024_0000h	144	BP_AGPIO144_ShutDown_L_UART0_INTR
GPIOx244	0000_0000h	145	BP_I2C0_SCL_EGPIO145_I3C0_SCL
GPIOx248	0006_0000h	146	BP_I2C0_SDA_EGPIO146_I3C0_SDA
GPIOx24C	0000_0000h	147	BP_I2C1_SCL_EGPIO147_I3C1_SCL
GPIOx250	0006_0000h	148	BP_I2C1_SDA_EGPIO148_I3C1_SDA
GPIOx254	0000_0000h	149	Reserved
GPIOx258	0000_0000h	150	Reserved
GPIOx25C	0000_0000h	151	Reserved
GPIOx260	0000_0000h	152	Reserved
GPIOx264	0024_0000h	153	BP_EGPIO153_UART4_CTS_L
GPIOx268	0014_0000h	154	BP_EGPIO154_UART4_RTS_L
GPIOx26C	0024_0000h	155	BP_EGPIO155_UART4_RXD
GPIOx270	0014_0000h	156	BP_EGPIO156_UART4_TXD
GPIOx274	0024_0000h	157	BP_AGPIO157_UART4_INTR
GPIOx278	0000_0000h	158	Reserved
GPIOx27C	0000_0000h	159	Reserved
GPIOx280	0000_0000h	160	Reserved
GPIOx284	0000_0000h	161	Reserved
GPIOx288	0000_0000h	162	Reserved
GPIOx28C	0000_0000h	163	Reserved
GPIOx290	0000_0000h	164	Reserved
GPIOx294	0000_0000h	165	Reserved
GPIOx298	0000_0000h	166	Reserved
GPIOx29C	0000_0000h	167	Reserved
GPIOx2A0	0000_0000h	168	Reserved
GPIOx2A4	0000_0000h	169	Reserved
GPIOx2A8	0000_0000h	170	Reserved
GPIOx2AC	0000_0000h	171	LPC_SPI_INT
GPIOx2B0	0000_0000h	172	Int_NBGppPmePulse
GPIOx2B4	0000_0000h	173	Int_NBGppHPPIpulse
GPIOx2B8	0000_0000h	174	Int_AcpiPerfIntr
GPIOx2BC	0000_0000h	175	Int_sata_sci_sci2_
GPIOx2C0	0000_0000h	176	Int_FanThermal_SCIOut
GPIOx2C4	0000_0000h	177	Int_ASFMasterIntr

GPIOx2C8	0000_0000h	178	Int_Ras_event
GPIOx2CC	0000_0000h	179	Int_GBL_RLS
GPIOx2D0	0000_0000h	180	Int_ShortTimerEvent_LongTimerEvent
GPIOx2D4	0000_0000h	181	Int_NBAssertion
GPIOx2D8	0000_0000h	182	Int_eSPIPme
GPIOx2DC	0000_0000h	183	Int_eSPISysEvt
GPIOx2E0	0000_0000h	184	Reserved
GPIOx2E4	0000_0000h	185	Reserved
GPIOx2E8	0000_0000h	186	Reserved
GPIOx2EC	0000_0000h	187	Reserved
GPIOx2F0	0000_0000h	188	GPIO Wake Status Index 0
GPIOx2F4	0000_0000h	189	GPIO Wake Status Index 1
GPIOx2F8	0000_0000h	190	GPIO Interrupt Status Index 0
GPIOx2FC	1f00_0000h	191	GPIO Interrupt Status Index 1

Table 107: GPIO BANK3 Register Default Value

Register	Reset Value	Index	Function
GPIOx00300	0000_0000h	192	
GPIOx00304	0000_0000h	193	
GPIOx00308	0000_0000h	194	
GPIOx0030C	0000_0000h	195	
GPIOx00310	0000_0000h	196	
GPIOx00314	0000_0000h	197	
GPIOx00318	0000_0000h	198	
GPIOx0031C	0000_0000h	199	
GPIOx00320	0000_0000h	200	
GPIOx00324	0000_0000h	201	
GPIOx00328	0000_0000h	202	
GPIOx0032C	0000_0000h	203	
GPIOx00330	0000_0000h	204	
GPIOx00334	0000_0000h	205	
GPIOx00338	0000_0000h	206	
GPIOx0033C	0000_0000h	207	
GPIOx00340	0000_0000h	208	
GPIOx00344	0000_0000h	209	
GPIOx00348	0000_0000h	210	
GPIOx0034C	0000_0000h	211	
GPIOx00350	0000_0000h	212	
GPIOx00354	0000_0000h	213	
GPIOx00358	0000_0000h	214	
GPIOx0035C	0000_0000h	215	
GPIOx00360	0000_0000h	216	
GPIOx00364	0000_0000h	217	
GPIOx00368	0000_0000h	218	
GPIOx0036C	0000_0000h	219	
GPIOx00370	0000_0000h	220	
GPIOx00374	0000_0000h	221	
GPIOx00378	0000_0000h	222	

GPIOx0037C	0000_0000h	223	
GPIOx00380	0000_0000h	224	
GPIOx00384	0000_0000h	225	
GPIOx00388	0000_0000h	226	
GPIOx0038C	0000_0000h	227	
GPIOx00390	0000_0000h	228	
GPIOx00394	0000_0000h	229	
GPIOx00398	0000_0000h	230	
GPIOx0039C	0000_0000h	231	
GPIOx003A0	0000_0000h	232	
GPIOx003A4	0000_0000h	233	
GPIOx003A8	0000_0000h	234	
GPIOx003AC	0000_0000h	235	
GPIOx003B0	0000_0000h	236	
GPIOx003B4	0000_0000h	237	
GPIOx003B8	0000_0000h	238	
GPIOx003BC	0000_0000h	239	
GPIOx003C0	0000_0000h	240	
GPIOx003C4	0000_0000h	241	
GPIOx003C8	0000_0000h	242	
GPIOx003CC	0000_0000h	243	
GPIOx003D0	0000_0000h	244	
GPIOx003D4	0000_0000h	245	
GPIOx003D8	0000_0000h	246	
GPIOx003DC	0000_0000h	247	
GPIOx003E0	0000_0000h	248	
GPIOx003E4	0000_0000h	249	
GPIOx003E8	0000_0000h	250	
GPIOx003EC	0000_0000h	251	
GPIOx003F0	0000_0000h	252	
GPIOx003F4	0000_0000h	253	
GPIOx003F8	0000_0000h	254	
GPIOx003FC	0000_0000h	255	

Table 108: Debounce Timer Definition

DebounceTmrLarge	DebounceTmrOutUnit	Timer Unit	Max Debounce Time
0	0	61 usec (2 RtcClk)	976 usec
0	1	244 usec (8 RtcClk)	3.9 msec
1	0	15.6 msec (512 RtcClk)	250 msec
1	1	62.5 msec (2048 RtcClk)	1.0 sec

GPIOx00000000 (FCH::GPIO::pwr_btn_l_agpio0)	
Reset: 0014_0000h.	
_aliasHOST; GPIOx00000000; GPIO=FED8_1500h	
Bits	Description
31	<b>less10secsts.</b> Read-only. Reset: 0. <b>Description:</b> This bit is only valid for GPIO0. For other GPIO, this bit is Reserved. When power button is pressed for less than 10 second in S0 state, this bit will become 1. This bit can be cleared by writing 1 to InterruptSts bit.
30	<b>less2secsts.</b> Read-only. Reset: 0. <b>Description:</b> This bit is only valid for GPIO0. For other GPIO, this bit is Reserved. When power button is pressed for less than 2 second in S0 state, this bit will become 1. When Less2secSts becomes 1, Less10secSts will also become 1. This bit can be cleared by writing 1 to InterruptSts bit.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h.

	<b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrmlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)



**GPIOx00000004 (FCH::GPIO::rst\_strap\_sys\_reset\_l\_agpio1)**

Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000004; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

GPIOx00000008 (FCH::GPIO::wake_1_agpio2)	
Reset: 0014_0000h.	
_aliasHOST; GPIOx00000008; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

GPIOx0000000C (FCH::GPIO::agpio3)	
Reset: 0014_0000h.	
_aliasHOST; GPIOx0000000C; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

GPIOx00000010 (FCH::GPIO::agpio4)	
Reset: 0024_0000h.	
_aliasHOST; GPIOx00000010; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)



**GPIOx00000014 (FCH::GPIO::agpio5)**

Reset: 0024\_0000h.

\_aliasHOST; GPIOx00000014; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx00000018 (FCH::GPIO::agpio6)**

Reset: 0024\_0000h.

\_aliasHOST; GPIOx00000018; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

GPIOx0000001C (FCH::GPIO::agpio7)	
Reset: 0014_0000h.	
_aliasHOST; GPIOx0000001C; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

GPIOx00000020 (FCH::GPIO::agpio8_tmu_clk_out0_tmu_clk_out1)	
Reset: 0024_0000h.	
_aliasHOST; GPIOx00000020; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)



GPIOx00000024 (FCH::GPIO::agpio9)	
Reset: 0024_0000h.	
_aliasHOST; GPIOx00000024; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

GPIOx00000028 (FCH::GPIO::agpio10_s0a3_gpio)	
Reset: 0014_0000h.	
_aliasHOST; GPIOx00000028; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

GPIOx0000002C (FCH::GPIO::agpio11_blink)	
Reset: 0014_0000h.	
_aliasHOST; GPIOx0000002C; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx00000030 (FCH::GPIO::llb\_l\_agpio12)**

Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000030; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)



GPIOx00000040 (FCH::GPIO::usb_oc0_1_agpio16)	
Reset: 0014_0000h.	
_aliasHOST; GPIOx00000040; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx00000044 (FCH::GPIO::usb\_oc1\_l\_agpio17)**

Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000044; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

GPIOx00000048 (FCH::GPIO::usb_oc2_l_agpio18)	
Reset: 0014_0000h.	
_aliasHOST; GPIOx00000048; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx0000004C (FCH::GPIO::smbus1\_scl\_agpio19\_i2c3\_scl\_i3c3\_scl)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx0000004C; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 0h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h.

	<b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)



**GPIOx00000050 (FCH::GPIO::smbus1\_sda\_agpio20\_i2c3\_sda\_i3c3\_sda)**

Reset: 0006\_0000h.

\_aliasHOST; GPIOx00000050; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 3h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h.

	<b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx00000054 (FCH::GPIO::espi\_reset\_l\_agpio21\_kbrst\_l)**

Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000054; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx00000058 (FCH::GPIO::espi\_alert\_1\_espi\_io1\_agpio22\_espi\_alert\_d1\_sd0\_cmd)**

Reset: 0024\_0000h.

\_aliasHOST; GPIOx00000058; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

GPIOx0000005C (FCH::GPIO::ac_pres_agpio23)	
Reset: 0014_0000h.	
_aliasHOST; GPIOx0000005C; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)



**GPIOx00000060 (FCH::GPIO::usb\_oc3\_l\_agpio24)**

Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000060; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

GPIOx00000068 (FCH::GPIO::pcie_rst0_l_egpio26)	
Read-write. Reset: 0024_0000h.	
_aliasHOST; GPIOx00000068; GPIO=FED8_1500h	
Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

GPIOx0000006C (FCH::GPIO::agpio27_pcie_rst1_l)	
Reset: 0024_0000h.	
_aliasHOST; GPIOx0000006C; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx00000074 (FCH::GPIO::spi\_tpm\_cs\_1\_agpio29)**

Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000074; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx00000078 (FCH::GPIO::spi\_cs2\_l\_agpio30\_espi\_cs\_l)**

Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000078; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)



15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx0000007C (FCH::GPIO::spi\_cs3\_l\_agpio31)**

Reset: 0014\_0000h.

\_aliasHOST; GPIOx0000007C; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx00000080 (FCH::GPIO::agpio32\_lpc\_rst\_l)**

Reset: 0024\_0000h.

\_aliasHOST; GPIOx00000080; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx00000098 (FCH::GPIO::clk\_req5\_l\_agpio38)**

Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000098; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

GPIOx0000009C (FCH::GPIO::clk_req6_1_agpio39)	
Reset: 0014_0000h.	
_aliasHOST; GPIOx0000009C; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)



15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

GPIOx000000A0 (FCH::GPIO::agpio40)	
Reset: 0024_0000h.	
_aliasHOST; GPIOx000000A0; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx000000A8 (FCH::GPIO::agpio42)**

Reset: 0014\_0000h.

\_aliasHOST; GPIOx000000A8; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx000000AC (FCH::GPIO::int\_fakests0)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000000AC; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000000B0 (FCH::GPIO::int\_shdwsysalarmfire)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000000B0; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

GPIOx000000B4 (FCH::GPIO::int_pwr_btn)	
Reset: 0000_0000h.	
_aliasHOST; GPIOx000000B4; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0.



	<b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx000000B8 (FCH::GPIO::int\_fakests0\_2nd)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000000B8; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000000BC (FCH::GPIO::int\_wake\_up\_wdt0)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000000BC; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000000C0 (FCH::GPIO::int\_dhcpleasetimerexpire)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000000C0; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000000C4 (FCH::GPIO::int\_asfslaveintr)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000000C4; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000000C8 (FCH::GPIO::int\_sm\_irq)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000000C8; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000000CC (FCH::GPIO::int\_wakefromllb)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000000CC; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

GPIOx000000D0 (FCH::GPIO::int_acdctimerevent)	
Reset: 0000_0000h.	
_aliasHOST; GPIOx000000D0; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000000D4 (FCH::GPIO::int\_althpet\_timersts)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000000D4; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.



**GPIOx000000D8 (FCH::GPIO::int\_isoevent0)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000000D8; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

GPIOx000000DC (FCH::GPIO::int_isocevent1)	
Reset: 0000_0000h.	
_aliasHOST; GPIOx000000DC; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

GPIOx000000E0 (FCH::GPIO::int_isocevent2)	
Reset: 0000_0000h.	
_aliasHOST; GPIOx000000E0; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000000E4 (FCH::GPIO::int\_isocevent3)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000000E4; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000000E8 (FCH::GPIO::int\_usb\_xhc\_0\_acpi\_pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000000E8; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000000EC (FCH::GPIO::int\_usb\_xhc\_1\_acpi\_pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000000EC; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000000F0 (FCH::GPIO::int\_rtc\_sts\_reg)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000000F0; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000000F4 (FCH::GPIO::int\_acp\_fch\_az\_wake)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000000F4; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.



**GPIOx000000F8 (FCH::GPIO::int\_acp\_fch\_i2s\_wake)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000000F8; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

GPIOx000000FC (FCH::GPIO::gpio_wake_interrupt_master_switch)	
Read-write. Reset: FF00_0000h.	
_aliasHOST; GPIOx000000FC; GPIO=FED8_1500h	
Bits	Description
31	<b>gpiowakeen.</b> Read-write. Reset: 1. When this bit is 0, all GPIO wake are blocked.
30	<b>gpiointerrupten.</b> Read-write. Reset: 1. When this bit is 0, all GPIO interrupts are blocked.
29	<b>eoi.</b> Read-write. Reset: 1. <b>Description:</b> This bit is set to 1 by SW when it is allowed to send GPIO interrupt. HW clear this bit when interrupt occurs. When this bit is 0, the GPIO interrupt is blocked. (This SW/HW handshake mechanism is the same with EOS of SMI.)
28	<b>mask_sts_en.</b> Read-write. Reset: 1. This bit enables HW to block all wake/intr status generation when SW writes any Debounce* registers. The length of blocking depends on mask_sts_length[3:0].
27:24	<b>mask_sts_length_3_0.</b> Read-write. Reset: Fh. <b>Description:</b> See mask_sts_en as well. The length of blocking = {mask_sts_length[11:0], 14'h3FFF}
23:16	<b>mask_sts_length_11_4.</b> Read-write. Reset: 00h. <b>Description:</b> See mask_sts_en as well. The length of blocking = {mask_sts_length[11:0], 14'h3FFF}
15	<b>enwinbluebtn.</b> Read-write. Reset: 0. <b>Description:</b> 0: GPIO0 detect debounced power button. Power button override is 4 sec. 1: GPIO0 detect debounced power button in S3/S5/S0i3, and detect "pressed less than 2sec" and "pressed 2~10sec" in S0. Power button override is 10 sec.
14	<b>introutactivehi.</b> Read-write. Reset: 0. <b>Description:</b> 0: GPIO controller interrupt output is low active 1: GPIO controller interrupt output is high active
13	<b>selgpio0src.</b> Read-write. Reset: 0. <b>Description:</b> Select the source for GPIO0 detection. 0: Power button goes to a processing logic first and then goes to GPIO0 detection logic 1: Power button goes to GPIO0 debounce and then goes to GPIO0 detection logic. Note: The "processing logic" includes 16ms debounce counter and a logic to detect how long the button has been pressed to generate press_less2s_sts and press_less4s_sts. "GPIO0 debounce block" only has debounce function.
12	<b>introutpulse.</b> Read-write. Reset: 0. <b>Description:</b> 0: GPIO controller interrupt output is a level signal 1: GPIO controller interrupt output is pulse signal Note: The polarity is defined by IntrOutActiveHi register bit.
11:0	Reserved.

**GPIOx0000010C (FCH::GPIO::spi\_rom\_req\_egpio67)**

Read-write. Reset: 0024\_0000h.

\_aliasHOST; GPIOx0000010C; GPIO=FED8\_1500h

Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

GPIOx00000110 (FCH::GPIO::spi1_dat2_agpio68_sd0_data3)	
Reset: 0014_0000h.	
_aliasHOST; GPIOx00000110; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

GPIOx00000114 (FCH::GPIO::spi1_dat3_agpio69_sd0_clk)	
Reset: 0014_0000h.	
_aliasHOST; GPIOx00000114; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx00000118 (FCH::GPIO::spi2\_clk\_egpio70)**

Read-write. Reset: 0024\_0000h.

\_aliasHOST; GPIOx00000118; GPIO=FED8\_1500h

Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.



GPIOx00000128 (FCH::GPIO::spi1_cs1_l_egpio74_gfx10_cac_ipio0)	
Read-write. Reset: 0014_0000h.	
_aliasHOST; GPIOx00000128; GPIO=FED8_1500h	
Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

GPIOx0000012C (FCH::GPIO::spi2_cs1_l_egpio75)	
Read-write. Reset: 0014_0000h.	
_aliasHOST; GPIOx0000012C; GPIO=FED8_1500h	
Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

**GPIOx00000130 (FCH::GPIO::spi\_rom\_gnt\_egpio76)**

Read-write. Reset: 0024\_0000h.

\_aliasHOST; GPIOx00000130; GPIO=FED8\_1500h

Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

**GPIOx00000134 (FCH::GPIO::spi1\_clk\_egpio77\_sd0\_data0)**

Read-write. Reset: 0024\_0000h.

\_aliasHOST; GPIOx00000134; GPIO=FED8\_1500h

Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

**GPIOx00000138 (FCH::GPIO::spi1\_cs2\_l\_egpio78\_gfx10\_cac\_ipio1\_sd0\_data1)**

Read-write. Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000138; GPIO=FED8\_1500h

Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

**GPIOx0000013C (FCH::GPIO::spi1\_cs3\_l\_egpio79)**

Read-write. Reset: 0014\_0000h.

\_aliasHOST; GPIOx0000013C; GPIO=FED8\_1500h

Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

**GPIOx00000140 (FCH::GPIO::spi1\_dat1\_egpio80\_sd0\_data2)**

Read-write. Reset: 0024\_0000h.

\_aliasHOST; GPIOx00000140; GPIO=FED8\_1500h

Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

GPIOx00000144 (FCH::GPIO::spi1_dat0_egpio81)	
Read-write. Reset: 0024_0000h.	
_aliasHOST; GPIOx00000144; GPIO=FED8_1500h	
Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.



**GPIOx00000150 (FCH::GPIO::fanin0\_agpio84)**

Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000150; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx00000154 (FCH::GPIO::intb\_fanout0\_fanout0\_agpio85)**

Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000154; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

GPIOx00000158 (FCH::GPIO::pkg_strap2_agpio86)	
Reset: 0014_0000h.	
_aliasHOST; GPIOx00000158; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

GPIOx00000164 (FCH::GPIO::genint1_l_agpio89_psp_intr0)	
Reset: 0014_0000h.	
_aliasHOST; GPIOx00000164; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)



**GPIOx00000168 (FCH::GPIO::genint2\_1\_agpio90\_psp\_intr1)**

Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000168; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

GPIOx0000016C (FCH::GPIO::spkr_agpio91)	
Reset: 0024_0000h.	
_aliasHOST; GPIOx0000016C; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx00000170 (FCH::GPIO::clk\_req0\_1\_agpio92)**

Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000170; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

GPIOx000001A0 (FCH::GPIO::spi2_dat0_egpio104)	
Read-write. Reset: 0024_0000h.	
_aliasHOST; GPIOx000001A0; GPIO=FED8_1500h	
Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

GPIOx000001A4 (FCH::GPIO::spi2_dat1_egpio105)	
Read-write. Reset: 0024_0000h.	
_aliasHOST; GPIOx000001A4; GPIO=FED8_1500h	
Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.



GPIOx000001A8 (FCH::GPIO::spi2_dat2_egpio106)	
Read-write. Reset: 0014_0000h.	
_aliasHOST; GPIOx000001A8; GPIO=FED8_1500h	
Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

GPIOx000001AC (FCH::GPIO::spi2_dat3_egpio107)	
Read-write. Reset: 0014_0000h.	
_aliasHOST; GPIOx000001AC; GPIO=FED8_1500h	
Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

**GPIOx000001C4 (FCH::GPIO::smbus0\_scl\_egpio113\_i2c2\_scl\_i3c2\_scl)**

Read-write. Reset: 0010\_0000h.

\_aliasHOST; GPIOx000001C4; GPIO=FED8\_1500h

Bits	Description
31:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 0h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

GPIOx000001C8 (FCH::GPIO::smbus0_sda_egpio114_i2c2_sda_i3c2_sda)	
Read-write. Reset: 0016_0000h.	
_aliasHOST; GPIOx000001C8; GPIO=FED8_1500h	
Bits	Description
31:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 3h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

GPIOx000001CC (FCH::GPIO::clk_req1_1_agpio115)	
Reset: 0014_0000h.	
_aliasHOST; GPIOx000001CC; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx000001D0 (FCH::GPIO::clk\_req2\_1\_agpio116)**

Reset: 0014\_0000h.

\_aliasHOST; GPIOx000001D0; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)



**GPIOx00000208 (FCH::GPIO::agpio130)**

Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000208; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx0000020C (FCH::GPIO::clk\_req3\_1\_egpio131)**

Read-write. Reset: 0014\_0000h.

\_aliasHOST; GPIOx0000020C; GPIO=FED8\_1500h

Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

**GPIOx00000210 (FCH::GPIO::clk\_req4\_l\_egpio132\_oscin)**

Read-write. Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000210; GPIO=FED8\_1500h

Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

GPIOx0000021C (FCH::GPIO::egpio135_uart2_cts_l_uart3_txd)	
Read-write. Reset: 0024_0000h.	
_aliasHOST; GPIOx0000021C; GPIO=FED8_1500h	
Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

**GPIOx00000220 (FCH::GPIO::egpio136\_uart2\_rxd)**

Read-write. Reset: 0024\_0000h.

\_aliasHOST; GPIOx00000220; GPIO=FED8\_1500h

Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

**GPIOx00000224 (FCH::GPIO::egpio137\_uart2\_rts\_l\_uart3\_rxd)**

Read-write. Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000224; GPIO=FED8\_1500h

Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

**GPIOx00000228 (FCH::GPIO::egpio138\_uart2\_txd)**

Read-write. Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000228; GPIO=FED8\_1500h

Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.



GPIOx0000022C (FCH::GPIO::agpio139_uart2_intr)	
Reset: 0024_0000h.	
_aliasHOST; GPIOx0000022C; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx00000230 (FCH::GPIO::egpio140\_uart0\_cts\_l\_uart1\_txd)**

Read-write. Reset: 0024\_0000h.

\_aliasHOST; GPIOx00000230; GPIO=FED8\_1500h

Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

GPIOx00000234 (FCH::GPIO::egpio141_uart0_rxd)	
Read-write. Reset: 0024_0000h.	
_aliasHOST; GPIOx00000234; GPIO=FED8_1500h	
Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

**GPIOx00000238 (FCH::GPIO::egpio142\_uart0\_rts\_l\_uart1\_rxd)**

Read-write. Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000238; GPIO=FED8\_1500h

Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

**GPIOx0000023C (FCH::GPIO::egpio143\_uart0\_txd)**

Read-write. Reset: 0014\_0000h.

\_aliasHOST; GPIOx0000023C; GPIO=FED8\_1500h

Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

**GPIOx00000240 (FCH::GPIO::agpio144\_shutdown\_1\_uart0\_intr)**

Reset: 0024\_0000h.

\_aliasHOST; GPIOx00000240; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)



**GPIOx00000244 (FCH::GPIO::i2c0\_scl\_egpio145\_i3c0\_scl)**

Read-write. Reset: 0010\_0000h.

\_aliasHOST; GPIOx00000244; GPIO=FED8\_1500h

Bits	Description
31:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 0h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

**GPIOx00000248 (FCH::GPIO::i2c0\_sda\_egpio146\_i3c0\_sda)**

Read-write. Reset: 0016\_0000h.

\_aliasHOST; GPIOx00000248; GPIO=FED8\_1500h

Bits	Description
31:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 3h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

**GPIOx0000024C (FCH::GPIO::i2c1\_scl\_egpio147\_i3c1\_scl)**

Read-write. Reset: 0010\_0000h.

\_aliasHOST; GPIOx0000024C; GPIO=FED8\_1500h

Bits	Description
31:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 0h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

**GPIOx00000250 (FCH::GPIO::i2c1\_sda\_egpio148\_i3c1\_sda)**

Read-write. Reset: 0016\_0000h.

\_aliasHOST; GPIOx00000250; GPIO=FED8\_1500h

Bits	Description
31:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 3h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

**GPIOx00000264 (FCH::GPIO::egpio153\_uart4\_cts\_l)**

Read-write. Reset: 0024\_0000h.

\_aliasHOST; GPIOx00000264; GPIO=FED8\_1500h

Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

**GPIOx00000268 (FCH::GPIO::egpio154\_uart4\_rts\_l)**

Read-write. Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000268; GPIO=FED8\_1500h

Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

GPIOx0000026C (FCH::GPIO::egpio155_uart4_rxd)	
Read-write. Reset: 0024_0000h.	
_aliasHOST; GPIOx0000026C; GPIO=FED8_1500h	
Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.

**GPIOx00000270 (FCH::GPIO::egpio156\_uart4\_txd)**

Read-write. Reset: 0014\_0000h.

\_aliasHOST; GPIOx00000270; GPIO=FED8\_1500h

Bits	Description
31:27	Reserved.
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25:24	Reserved.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:0	Reserved.



**GPIOx00000274 (FCH::GPIO::agpio157\_uart4\_intr)**

Reset: 0024\_0000h.

\_aliasHOST; GPIOx00000274; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	<b>rxenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable GPIO PAD receive 1: Enable GPIO PAD receive ( Default ENABLE. Insert inverter already, so keep default value as 1'b0 )
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23	<b>outputenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Output is disabled on the pin 1: Output is enabled on the pin.
22	<b>outputvalue.</b> Read-write. Reset: 0. <b>Description:</b> 0: low. 1: high.
21	<b>pulldownenable.</b> Read-write. Reset: 1. <b>Description:</b> 0: Pull-down is disabled on the pin. 1: Pull-down is enabled on the pin.
20	<b>pullupenable.</b> Read-write. Reset: 0. <b>Description:</b> 0: Pull-up is disabled on the pin. 1: Pull-up is enabled on the pin.
19	Reserved.
18:17	<b>drvstrengthsel.</b> Read-write. Reset: 2h. <b>Description:</b> 3.3V PAD: x0: Z= 40 ohms x1: Z= 80 ohms 1.8V PAD: 00: not supported 01: Z= 60 ohms 10: Z= 40 ohms 11: Z= 80 ohms
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)

15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7	<b>debouncetmrlarge.</b> Read-write. Reset: 0. Combined with DebounceTmrOutUnit, this bit change the unit and max debounce time for the debounce timer. Please refer to "DebounceTmrOutUnit" description for details.
6:5	<b>debouncecntrl.</b> Read-write. Reset: 0h. <b>Description:</b> 00: No debounce 01: Preserve low glitch 10: Preserve high glitch 11: Remove glitch (See Note below)
4	<b>debouncetmroutunit.</b> Read-write. Reset: 0. <b>Description:</b> {DebounceTmrLarge, DebounceTmrOutUnit} defines the unit and max debounce time for the debounce timer, 00: Timer Unit = 61 usec (2 RtcClk), Max Debounce Time = 976 usec 01: Timer Unit = 244 usec (8 RtcClk), Max Debounce Time = 3.9 msec 10: Timer Unit = 15.6 msec (512 RtcClk), Max Debounce Time = 250 msec 11: Timer Unit = 62.5 msec (2048 RtcClk), Max Debounce Time = 1.0 sec
3:0	<b>debouncetmrout.</b> Read-write. Reset: 0h. Debounce timer out number ( 0 means that debouncing logic is disabled)

**GPIOx000002AC (FCH::GPIO::lpc\_spi\_int)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000002AC; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000002B0 (FCH::GPIO::int\_nbgpppmepulse)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000002B0; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000002B4 (FCH::GPIO::int\_nbgpphpulse)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000002B4; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000002B8 (FCH::GPIO::int\_acpiperfintr)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000002B8; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000002BC (FCH::GPIO::int\_sata\_sci\_sci2)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000002BC; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

GPIOx000002C0 (FCH::GPIO::int_fanthermal_sciout)	
Reset: 0000_0000h.	
_aliasHOST; GPIOx000002C0; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.



**GPIOx000002C4 (FCH::GPIO::int\_asfmasterintr)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000002C4; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000002C8 (FCH::GPIO::int\_ras\_event)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000002C8; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000002CC (FCH::GPIO::int\_gbl\_rls)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000002CC; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

GPIOx000002D0 (FCH::GPIO::int_shorttimerevent_longtimerevent)	
Reset: 0000_0000h.	
_aliasHOST; GPIOx000002D0; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000002D4 (FCH::GPIO::int\_nbassertion)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx000002D4; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

GPIOx000002D8 (FCH::GPIO::int_espipme)	
Reset: 0001_0000h.	
_aliasHOST; GPIOx000002D8; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 1. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic) (NOTE: This Event is low enable)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000002DC (FCH::GPIO::int\_espisysevt)**

Reset: 0001\_0000h.

\_aliasHOST; GPIOx000002DC; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 1. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic) (NOTE: This Event is low enable)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx000002F0 (FCH::GPIO::gpio\_wake\_status\_index\_0)**

Read-only. Reset: 0000\_0000h.

\_aliasHOST; GPIOx000002F0; GPIO=FED8\_1500h

Bits	Description
31:16	<b>wake_status_index_15_0.</b> Read-only. Reset: 0000h. When this bit is 1, it means at least one of the wake status of GPIO N*4 ~ N*4+3 is 1. N = 16~31
15:0	<b>wake_status_index_s5_15_0.</b> Read-only. Reset: 0000h. When this bit is 1, it means at least one of the wake status of GPIO N*4 ~ N*4+3 is 1. N = 0~15

**GPIOx000002F4 (FCH::GPIO::gpio\_wake\_status\_index\_1)**

Read-only. Reset: 0000\_0000h.

\_aliasHOST; GPIOx000002F4; GPIO=FED8\_1500h

Bits	Description
31:16	Reserved.
15	<b>nbgpppmewake.</b> Read-only. Reset: 0. <b>Description:</b> 1: NB Gpp has sent PME to wake the system. (one of wake status is set in Gpio Bank3 register) 0: no NB Gpp Pme wake event
14:0	<b>wake_status_index_30_16.</b> Read-only. Reset: 0000h. When this bit is 1, it means at least one of the wake status of GPIO N*4 ~ N*4+3 is 1. N = 32~46

**GPIOx000002F8 (FCH::GPIO::gpio\_interrupt\_status\_index\_0)**

Read-only. Reset: 0000\_0000h.

\_aliasHOST; GPIOx000002F8; GPIO=FED8\_1500h

Bits	Description
31:16	<b>interrupt_status_index_15_0.</b> Read-only. Reset: 0000h. When this bit is 1, it means at least one of the interrupt status of GPIO N*4 ~ N*4+3 is 1. N = 16~31
15:0	<b>interrupt_status_index_s5_15_0.</b> Read-only. Reset: 0000h. When this bit is 1, it means at least one of the interrupt status of GPIO N*4 ~ N*4+3 is 1. N = 0~15

**GPIOx000002FC (FCH::GPIO::gpio\_interrupt\_status\_index\_1)**

Reset: 1F00\_0000h.

\_aliasHOST; GPIOx000002FC; GPIO=FED8\_1500h

Bits	Description
31:29	Reserved.
28	<b>mask_sts_en.</b> Read-write. Reset: 1. This bit enables HW to block all wake/intr status generation when SW writes any Debounce* registers. The length of blocking depends on mask_sts_length[3:0].
27:24	<b>mask_sts_length_3_0.</b> Read-write. Reset: Fh. <b>Description:</b> See mask_sts_en as well. The length of blocking = {mask_sts_length[11:0], 14'h3FFF}
23:16	<b>mask_sts_length_11_4.</b> Read-write. Reset: 00h. <b>Description:</b> See mask_sts_en as well. The length of blocking = {mask_sts_length[11:0], 14'h3FFF}
15	<b>nbgpppmeintr.</b> Read-only. Reset: 0. <b>Description:</b> 1: NB GPP has sent PME (one of Intr status is set in Gpio Bank3 register) 0: no NB GPP Pme event
14:0	<b>interrupt_status_index_30_16.</b> Read-only. Reset: 0000h. When this bit is 1, it means at least one of the interrupt status of GPIO N*4 ~ N*4+3 is 1. N = 32~46



**GPIOx00000300 (FCH::GPIO::int\_nbgppportdev0pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000300; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx00000304 (FCH::GPIO::int\_nbgppportdev1pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000304; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx00000308 (FCH::GPIO::int\_nbgppportdev2pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000308; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx0000030C (FCH::GPIO::int\_nbgppportdev3pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx0000030C; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx00000310 (FCH::GPIO::int\_nbgppportdev4pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000310; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

GPIOx00000314 (FCH::GPIO::int_nbgppportdev5pme)	
Reset: 0000_0000h.	
_aliasHOST; GPIOx00000314; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx00000318 (FCH::GPIO::int\_nbgppportdev6pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000318; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx0000031C (FCH::GPIO::int\_nbgppportdev7pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx0000031C; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.



**GPIOx00000320 (FCH::GPIO::int\_nbgppportdev8pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000320; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

GPIOx00000324 (FCH::GPIO::int_nbgppportdev9pme)	
Reset: 0000_0000h.	
_aliasHOST; GPIOx00000324; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx00000328 (FCH::GPIO::int\_nbgppportdev10pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000328; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx0000032C (FCH::GPIO::int\_nbgppportdev11pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx0000032C; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx00000330 (FCH::GPIO::int\_nbgppportdev12pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000330; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx00000334 (FCH::GPIO::int\_nbgppportdev13pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000334; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx00000338 (FCH::GPIO::int\_nbgppportdev14pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000338; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx0000033C (FCH::GPIO::int\_nbgppportdev15pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx0000033C; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.



**GPIOx00000340 (FCH::GPIO::int\_nbgppportdev16pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000340; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx00000344 (FCH::GPIO::int\_nbgppportdev17pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000344; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx00000348 (FCH::GPIO::int\_nbgppportdev18pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000348; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx0000034C (FCH::GPIO::int\_nbgppportdev19pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx0000034C; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx00000350 (FCH::GPIO::int\_nbgppportdev20pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000350; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx00000354 (FCH::GPIO::int\_nbgppportdev21pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000354; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx00000358 (FCH::GPIO::int\_nbgppportdev22pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000358; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx0000035C (FCH::GPIO::int\_nbgppportdev23pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx0000035C; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.



**GPIOx00000360 (FCH::GPIO::int\_nbgppportdev24pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000360; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx00000364 (FCH::GPIO::int\_nbgppportdev25pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000364; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

GPIOx00000368 (FCH::GPIO::int_nbgppportdev26pme)	
Reset: 0000_0000h.	
_aliasHOST; GPIOx00000368; GPIO=FED8_1500h	
Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx0000036C (FCH::GPIO::int\_nbgppportdev27pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx0000036C; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx00000370 (FCH::GPIO::int\_nbgppportdev28pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000370; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx00000374 (FCH::GPIO::int\_nbgppportdev29pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000374; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx00000378 (FCH::GPIO::int\_nbgppportdev30pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx00000378; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlcn.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlcn.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.

**GPIOx0000037C (FCH::GPIO::int\_nbgppportdev31pme)**

Reset: 0000\_0000h.

\_aliasHOST; GPIOx0000037C; GPIO=FED8\_1500h

Bits	Description
31:30	Reserved.
29	<b>wakests.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't generate wake event. 1: the pin is one of wake source Writing 1 to clear the bit to 0.
28	<b>interruptsts.</b> Read-write,Read,Write-1-to-clear. Reset: 0. <b>Description:</b> 0: the pin doesn't cause interrupt. 1: the pin is one of interrupt source Writing 1 to clear the bit to 0.
27	<b>wakecntrlz.</b> Read-write. Reset: 0. WakeCntrlZ enables wake in Z state
26	Reserved.
25	<b>swcntrlen.</b> Read-write. Reset: 0. <b>Description:</b> 0: Disable software controlled GPIO in 1: Enable software controlled GPIO in
24	<b>swcntrlin.</b> Read-write. Reset: 0. When SwCntrlEn is 1, this bit can be used by software to control the GPIO input to the GPIO controller. This might be useful during debug or work-around.
23:17	Reserved.
16	<b>pinsts.</b> Read-write. Reset: 0. <b>Description:</b> Read only. 0: the pin is low 1: the pin is high (NOTE: the bit doesn't affected by debounce logic)
15:13	<b>wakecntrl_2_0.</b> Read-write. Reset: 0h. <b>Description:</b> WakeCntrl[0] enables wake in S0i3 state WakeCntrl[1] enables wake in S3 state (SLP_TYP = 3, and !G0_State) WakeCntrl[2] enables wake in S4/S5 state (SLP_TYP = 4 or 5, and !G0_State)
12:11	<b>interruptenable_1_0.</b> Read-write. Reset: 0h. <b>Description:</b> InterruptEnable[0] enables interrupt status InterruptEnable[1] enables interrupt delivery
10:9	<b>activelevel.</b> Read-write. Reset: 0h. <b>Description:</b> 00: Active high 01: Active Low. 10: Active on both of edges if LevelTrig is set to 0 11: reserved
8	<b>leveltrig.</b> Read-write. Reset: 0. <b>Description:</b> 0: Edge trigger 1: Level trigger
7:0	Reserved.



## List of Namespaces

Namespace	Heading(s)
Core::X86::Apic	2.1.11.2.2 [Local APIC Registers]
Core::X86::Cpuid	2.1.12.1 [CPUID Instruction Functions]
Core::X86::Msr	2.1.13.1 [MSRs - MSR0000_xxxx] 2.1.13.2 [MSRs - MSRC000_xxxx] 2.1.13.3 [MSRs - MSRC001_0xxx] 2.1.13.4 [MSRs - MSRC001_1xxx]
Core::X86::Pmc::Core	2.1.14.4 [Large Increment per Cycle Events] 2.1.14.5.1 [Floating-Point (FP) Events] 2.1.14.5.2 [Load/Store (LS) Events] 2.1.14.5.3 [Instruction Cache (IC) and Branch Prediction (BP) Events] 2.1.14.5.4 [DE Events] 2.1.14.5.5 [EX (SC) Events]
Core::X86::Pmc::L2	2.1.14.5.6 [L2 Cache Events]
Core::X86::Pmc::L3	2.1.14.6.1 [L3 Cache PMC Events]
Core::X86::Smm	2.1.11.1.6 [System Management State]
FCH::AOAC	7.3.7 [Always On Always Connected (AOAC) Registers]
FCH::GPIO	7.3.10.2 [GPIO Registers]
FCH::IO	7.3.1.1 [Registers]
FCH::IOAPIC	7.3.2.1 [IOAPIC Registers]
FCH::IOMUX	7.3.10.1.2 [IOMUX Funtional Table]
FCH::ITF::ESPI	7.3.8.1 [eSPI Registers]
FCH::MISC	7.3.9.1 [Miscellaneous (MISC) Registers]
FCH::PM	7.3.9.2 [Power Management (PM) Registers and Standard ACPI Registers]
FCH::PM2	7.3.9.4 [Power Management (PM2) Registers]
FCH::PM::RTCEXT	7.3.9.3 [RTC External Registers]
FCH::SMI	7.3.3 [SMI Registers]
FCH::TMR::ACDC	7.3.6 [Wake Alarm Device (AcDcTimer) Registers]
FCH::TMR::HPET	7.3.4 [High Precision Event Timer (HPET) Registers]

FCH::TMR::WDT	7.3.5 [Watchdog Timer (WDT) Registers]
IO	2.1.7 [PCI Configuration Legacy Access]
L3::L3CRB	2.2.1 [L3 MSR Registers]
L3::L3CT	2.2.2 [L3 Clocks and Test (CT) MSR Registers.]
MCA::CS	3.2.5.8 [CS]
MCA::DE	3.2.5.4 [DE]
MCA::EX	3.2.5.5 [EX]
MCA::FP	3.2.5.6 [FP]
MCA::IF	3.2.5.2 [IF]
MCA::L2	3.2.5.3 [L2]
MCA::L3	3.2.5.7 [L3]
MCA::LS	3.2.5.1 [LS]
MCA::NBIO	3.2.5.11 [NBIO]
MCA::PIE	3.2.5.9 [PIE]
MCA::UMC	3.2.5.10 [UMC]
SBRMI	4.6 [SB-RMI Registers]
SBTSI	5.4 [SB-TSI Registers]

## List of Definitions

**ABS:** ABS(integer expression): Remove sign from signed value.

**AGESA™:** AMD Generic Encapsulated Software Architecture.

**AP:** Application Processor.

**APML:** Advanced Platform Management Link.

**APU:** Accelerated Processing Unit.

**ARA:** Alert response address.

**ARP:** Address Resolution Protocol

**BAR:** The BAR, or base address register, physical register mnemonic format is of the form PREFIXxZZZ. PREFIX is an all capital letter name that connotes the BAR to which the offset is added to get the physical address of the operation. ZZZ is the offset.

**BatteryPower:** The system is running from a limited energy or battery power source or otherwise undocked from a continuous power supply. Setting using this definition may be required to change during run time.

**BCD:** Binary Coded Decimal number format.

**BCS:** Base Configuration Space.

**BIST:** Built-In Self-Test. Hardware within the processor that generates test patterns and verifies that they are stored correctly (in the case of memories) or received without error (in the case of links).

**BMC:** Base management controller.

**Boot VID:** Boot Voltage ID. This is the VDD and VDDNB voltage level that the processor requests from the external voltage regulator during the initial phase of the cold boot sequence.

**BSC:** Boot strap core. Core 0 of the BSP.

**BSP:** Boot strap processor.

**C-states:** These are ACPI defined core power states. C0 is operational. All other C-states are low-power states in which the processor is not executing code. See docACPI.

**Canonical-address:** An address in which the state of the most-significant implemented bit is duplicated in all the remaining higher-order bits, up to bit[63].

**CCX:** Core Complex where more than one core shares L3 resources.

**CEIL:** CEIL(real expression): Rounds real number up to nearest integer.

**CMP:** Specifies the core number.

**COF:** Current operating frequency of a given clock domain.

**Cold reset:** PWROK is de-asserted and RESET\_L is asserted.

**Configurable:** Indicates that the access type is configurable as described by the documentation.

**CoreCOF:** Core current operating frequency in MHz. CoreCOF = Core::X86::Msr::PStateDef[CpuFid[11:0]] \* 5MHz.

**COUNT:** COUNT(integer expression): Returns the number of binary 1's in the integer.

**CpuCoreNum:** Specifies the core number.

**CPUID:** The CPUID, or x86 processor identification state, physical register mnemonic format is of the form CPUID FnXXXX\_XXXX\_EiX[\_xYYY], where XXXX\_XXXX is the hex value in the EAX and YYY is the hex value in ECX.

**DID:** Divisor Identifier. Specifies the post-PLL divisor used to reduce the COF.

**docACPI:** Advanced Configuration and Power Interface (ACPI) Specification. <http://uefi.org/specifications>.

**docAPM1:** AMD64 Architecture Programmer's Manual Volume 1: Application Programming, order# 24592.

**docAPM2:** AMD64 Architecture Programmer's Manual Volume 2: System Programming, order# 24593.

**docAPM3:** AMD64 Architecture Programmer's Manual Volume 3: Instruction-Set Reference, order# 24594.

**docAPM4:** AMD64 Architecture Programmer's Manual Volume 4: 128-Bit and 256-Bit Media Instructions, order# 26568.

**docAPM5:** AMD64 Architecture Programmer's Manual Volume 5: 64-Bit Media and x87 Floating-Point Instructions, order# 26569.

**docASF:** Alert Standard Format Specification. <http://dmtf.org/standards/asf>.

**docDP:** VESA DisplayPort Standard. <http://www.vesa.org/vesa-standards>.

**docI2C:** I2C Bus Specification.

[http://www.nxp.com/documents/user\\_manual/UM10204.pdf](http://www.nxp.com/documents/user_manual/UM10204.pdf)

**docIOMMU:** AMD I/O Virtualization Technology Specification, order# 48882.

**docJEDEC:** JEDEC Standards. <http://www.jedec.org>.

**docPCIe:** PCI Express® Specification. <http://www.pcisig.org>.

**docPCIb:** PCI Local Bus Specification. <http://www.pcisig.org>.

**docSDHC:** Secure Digital Host Controller Standard Specification.

<https://www.sdcard.org>.

**docUSB:** Universal Serial Bus Specification. <http://www.usb.org>.

**Doubleword:** A 32-bit value.

**DPIK:** PHY Interface De-Kompressor

**DW:** Doubleword.

**DXIO:** Distributed IO Crossbar Subsystem

**EC:** Embedded Controller.

**ECS:** Extended Configuration Space.

**EDC:** Electrical design current. Indicates the maximum current the voltage rail can demand for a short, thermally insignificant time.

**Error-on-read:** Error occurs on read.

**Error-on-write:** Error occurs on write.

**Error-on-write-0:** Error occurs on bitwise write of 0.

**Error-on-write-1:** Error occurs on bitwise write of 1.

**FCH:** The integrated platform subsystem that contains the IO interfaces and bridges them to the system BIOS. Previously included in the Southbridge.

**FID:** Frequency Identifier. Specifies the PLL frequency multiplier for a given clock domain.

**FLOOR:** FLOOR(integer expression): Rounds real number down to nearest integer.

**FP8:** Notebook package for direct solder boards (uBGA).

**FreeRunSampleTimer:** An internal free running timer used by many power management features.

**GOP:** GMI over PCIe® link, referred to as the data link layer of either xGMI or WAFL

**GOPX1:** GMI over PCIe® link, with a link-width of 1 lane, referred to as WAFL link layer

**GOPX16:** GMI over PCIe® link, with a link-width of 16 lanes, referred to as xGMI data link layer

**GT/s:** Giga-Transfers per second.

**HTC:** Hardware Thermal Control.

**HTC-active state:** Hardware-controlled lower-power, lower performance state used to reduce temperature.

**HWPF:** Hardware Prefetcher.

**IBS:** Instruction based sampling.

**IFCM:** Isochronous flow-control mode, as defined in the link specification.

**Inaccessible:** Not readable or writable (e.g., Hide ? Inaccessible : Read-Write).

**IO configuration:** Access to configuration space through IO ports CF8h and CFCh.

**IORR:** IO range register.

**KBC:** Keyboard Controller.

**KPFIFO:** KPRI Transmit Elasticity FIFO element

**KPMX:** KPRI multiplexing element

**KPNP:** Kompressed Packet Near PHY element

**KPRI:** Kompressed Physical-to-Raw PCS Interface

**L1 cache:** The level 1 caches (instruction cache and the data cache).

**L2 cache:** The level 2 caches.

**Linear (virtual) address:** The address generated by a core after the segment is applied.

**LINT:** Local interrupt.

**Logical address:** The address generated by a core before the segment is applied.

**logical mnemonic:** The register mnemonic format that describes the register functionally, what namespace to which the register belongs, a name for the register that connotes its function, and optionally, named parameters that indicate the different function of each instance (e.g., Link::Phy::PciDevVendIDF3). See 1.4.3.1 [Logical Mnemonic].

**LRU:** Least recently used.

**LVT:** Local vector table. A collection of APIC registers that define interrupts for local events (e.g., APIC[530:500] [Extended Interrupt [3:0] Local Vector Table]).

**Macro-op:** The front-end of the pipeline breaks instructions into macro-ops and transfers (dispatches) them to the back-end of the pipeline for scheduling and execution. See Software Optimization Guide.

**Master or SMBus Master:** The device that initiates and terminates all communication and drives the clock, SCL.

**MAX:** MAX(integer expression list): Picks maximum integer or real value of comma separated list.

**MB:** Megabyte; 1024 KB.

**MCA:** Machine Check Architecture.

**MCAX:** Machine Check Architecture eXtensions.

**MergeEvent:** A PMC event that is capable of counter increments greater than 15, thus requiring merging a pair of even/odd performance monitors.

**Micro-op:** Processor schedulers break down macro-ops into sequences of even simpler instructions called micro-ops, each of which specifies a single primitive operation. See Software Optimization Guide.

**MIN:** MIN(integer expression list): Picks minimum integer or real value of comma separated list.

**MMIO:** Memory-Mapped Input-Output range. This is physical address space that is mapped to the IO functions such as the IO links or MMIO configuration.

**MMIO configuration:** Access to configuration space through memory space.

**MPB:** Microcode patch block.

**MSR:** The MSR, or x86 model specific register, physical register mnemonic format is of the form MSRXXXX\_XXXX, where XXXX\_XXXX is the hexadecimal MSR number. This space is accessed through x86 defined RDMSR and WRMSR instructions.

**MTRR:** Memory-type range register. The MTRRs specify the type of memory associated with various memory ranges.

**NBC:** NBC=(CUID Fn00000001\_EBX[LocalApicId[3:0]] == 0). Node Base Core. The lowest numbered core in the node.

**NTA:** Non-Temporal Access.

**OW:** Octword. An 128-bit value.

**PCICFG:** The PCICFG, or PCI defined configuration space, physical register mnemonic format is of the form DXFYxZZZ. Bus 0 is implied, X specifies the hexadecimal device number (this may be 1 or 2 digits). Y specifies the function number. ZZZ specifies the hexadecimal byte address (this may be 2 or 3 digits). Example: D18F2x40 specifies the register at bus 0, device 18h, function 2, and address 40h. If the mnemonic starts with B, then the physical mnemonic format is BWDXFYxZZZ where WW specifies the hexadecimal bus number (1 or 2 hex digits) or "XX" implying that the bus is relocatable. Example; BXXD00F6x40 specifies that the bus is relocatable, B0AD00F2x000 specifies that the bus is 0Ah.

**PCIe@:** PCI Express.

**PCS:** Physical Coding Sublayer.

**PEC:** Packet error code.

**physical mnemonic:** The register mnemonic that is formed based on the physical address used to access the register (e.g., D18F3x00). See 1.4.3.2 [Physical Mnemonic].

**PIK:** PHY Interface Kompressor

**PMA:** Physical Media Access. a.k.a. PHY

**PMC:** The PMC, or x86 performance monitor counter, physical register mnemonic format is any of the forms {PMCxxx, L2IPMCxxx, NBPMCxxx}, where xxx is the performance monitor select.

**POR:** Power on reset.

**POW:** POW(base, exponent): POW(x,y) returns the value x to the power of y.

**PPIN:** Protected Processor Inventory Number.

**Processor:** A package containing one or more Nodes. See Node.

**PTE:** Page table entry.

**QW:** Quadword. A 64-bit value.

**REFCLK:** Reference clock. Refers to the clock frequency (100 MHz) or the clock period (10 ns) depending on the context used.

**register instance parameter specifier:** A register instance parameter specifier is of the form \_register parameter name[register parameter value list] (e.g., The register instance parameter specifier \_dct[1:0] has a register parameter name of dct (The DCT PHY instance name) and a register parameter value list of "1:0" or 2 instances of DCT PHY).

**register instance specifier:** The register instance specifier exists when there is more than one instance for a register. The register instance specifier consists of one or more register instance parameter specifier (e.g., The register instance specifier \_dct[1:0]\_chiplet[BCST,3:0]\_pad[BCST,11:0] consists of 3 register instance parameter specifiers, \_dct[1:0], \_chiplet[BCST,3:0], and \_pad[BCST,11:0]).

**register name:** A name that connotes the function of the register.

**register namespace:** A namespace for which the register name must be unique. A register namespace indicates to which IP it belongs and an IP may have multiple namespaces. A namespace is a string that supports a list of "::-" separated names. The convention is for the list of names to be hierarchical, with the most significant name first and the least significant name last (e.g., Link::Phy::Rx is the RX component in the Link PHY).

**register parameter name:** A register parameter name is the name of the number of instances at some level of the logical hierarchy (e.g., The register

parameter name dct specifies how many instances of the DCT PHY exist).

**register parameter value list:** The register parameter value list is the logical name for each instance of the register parameter name (e.g., For \_dct[1:0], there are 2 DCT PHY instances, with the logical names 0 and 1, but it should be noted that the logical names 0 and 1 can correspond to physical values other than 0 and 1). It is the purpose of the AddressMappingTable to map these register parameter values to physical address values for the register.

**Reserved-write-as-0:** Reads are undefined. Must always write 0.

**Reserved-write-as-1:** Reads are undefined. Must always write 1.

**ROUND:** ROUND(real expression): Rounds to the nearest integer; halfway rounds away from zero.

**RTS:** Remote temperature sensor, typical examples are ADM1032, LM99, MAX6657, EMC1002.

**SB-RMI:** Remote Management interface.

**SB-TSI:** Sideband Internal Temperature Sensor Interface. See APML.

**Shutdown:** A state in which the affected core waits for either INIT, RESET, or NMI. When shutdown state is entered, a shutdown special cycle is sent on the IO links.

**Slave or SMBus slave:** The slave cannot initiate SMBus communication and cannot drive the clock but can drive the data signal SDA and the alert signal ALERT\_L.

**SMAF:** System Management Action Field. This is the code passed from the SMC to the processors in STPCLK assertion messages.

**SMI:** System management interrupt.

**SMM:** System Management Mode.

**SMT:** Simultaneous multithreading. See Core::X86::CpuId::CoreId[ThreadsPerCore].

**Speculative event:** A performance monitor event counter that counts all occurrences of the event even if the event occurs during speculative code execution.

**SSC:** Spread Spectrum Clocking.

**SVM:** Secure virtual machine.

**TCC:** Temperature calculation circuit.

**Tctl:** Processor temperature control value.

**TDC:** Thermal Design Current.

**TDP:** Thermal Design Power. A power consumption parameter that is used in conjunction with thermal specifications to design appropriate cooling solutions for the processor.

**Thread:** One architectural context for instruction execution.

**TOM2:** Top of extended Memory.

**TSI:** Temperature sensor interface.

**TSM:** Temperature sensor macro.

**UMI:** Unified Media Interface. The link between the processor and the FCH.

**UNIT:** UNIT(register field reference): Input operand is a register field reference that contains a valid values table that defines a value with a unit (e.g., clocks, ns, ms, etc.). This function takes the value in the register field and returns the value associated with the unit (e.g., If the field had a valid value definition where 1010b was defined as 5 ns). Then if the field had the value of 1010b, then UNIT() would return the value 5.

**Unpredictable:** The behavior of both reads and writes is unpredictable.

**VID:** Voltage level identifier.

**VMPL:** Virtual Machine Privilege Level.

**Volatile:** Indicates that a register field value may be modified by hardware, firmware, or microcode when fetching the first instruction and/or might have read or write side effects. No read may depend on the results of a previous read and no write may be omitted based on the value of a previous read or write. Not volatile indicates that software may service a read from the results of a previous read and that a write may be dropped if its value matches the value previously read or written.

**Warm reset:** RESET\_L is asserted only (while PWROK stays high).

**WDT:** Watchdog timer. A timer that detects activity and triggers an error if a specified period of time expires without the activity.

**WRIG:** Writes Ignored.

**Write-0-only:** Writing a 0 clears to a 0; Writing a 1 has no effect. If not associated with Read, then reads are undefined.

**Write-1-only:** Writing a 1 sets to a 1; Writing a 0 has no effect. If not associated with Read, then reads are undefined.

**Write-1-to-clear:** Writing a 1 clears to a 0; Writing a 0 has no effect. If not associated with Read, then reads are undefined.

**Write-once:** Capable of being written once; all subsequent writes have no effect. If not associated with Read, then reads are undefined.

**X2APICEN:** x2 APIC is enabled. X2APICEN = (Core::X86::Msr::APIC\_BAR[ApicEn] &&

Core::X86::Msr::APIC\_BAR[x2ApicEn]).  
**XBAR**: Cross bar; command packet switch.