

**Processor Programming
Reference (PPR)
for AMD Family 1Ah
Model 11h, Revision B0
Processors
Volume 2 of 7**

Legal Notices

© 2022-2024 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.

AMD-V 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.

EPYC is a trademark of Advanced Micro Devices, Inc.

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.

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

PCIe is a registered trademark of PCI-SIG Corporation.

SoundWire is a registered trademark of MIPI Alliance, Inc.

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

Volume 1:

- 1** [Overview](#)
- 2** [Core Complex \(CCX\)](#)

Volume 2:

- 3** **Reliability, Availability, and Serviceability (RAS) Features**

Volume 3:

- 4** [System Management Unit \(SMU\)](#)

Volume 4:

- 5** [Advanced Platform Management Link \(APML\)](#)
- 6** [SB Temperature Sensor Interface \(SB-TSI\)](#)
- 7** [Host System Management Port \(HSMP\)](#)
- 8** [Data Fabric \(DF\)](#)
- 9** [Unified Memory Controller \(UMC\)](#)

Volume 5:

- 10** [Northbridge IO \(NBIO\)](#)

Volume 6:

- 11** [Fusion Controller Hub \(FCH\)](#)

Volume 7:

- 12** [Reserved](#)

[List of Namespaces](#)

[List of Definitions](#)

Table of Contents

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 PSP
- 3.2.5.12 SMU
- 3.2.5.13 MP5
- 3.2.5.14 NBIO
- 3.2.5.15 PCIE
- 3.2.5.16 KPX SERDES
- 3.2.5.17 KPX GMI
- 3.2.5.18 PCS GMI
- 3.2.5.19 PCS XGMI
- 3.2.5.20 NBIF
- 3.2.5.21 SHUB
- 3.2.5.22 USB
- 3.2.5.23 SATA
- 3.2.5.24 MPDMA
- 3.3 PCI Express® RAS Features
 - 3.3.1 PCI Express® Advanced Error Reporting (AER)
 - 3.3.1.1 Recommended PCI Express® AER Severity
 - 3.3.2 PCIe® Error Status
 - 3.3.3 PCI Express® eDPC
 - 3.3.4 PCI Express® ECRC

List of Figures

List of Tables

Table 31:	Machine Check Terms and Acronyms
Table 32:	Legacy MCA MSR Layout
Table 33:	MCAX MSR Layout
Table 34:	MCAX Implementation-Specific Register Layout
Table 35:	Error Overwrite Priorities
Table 36:	Error Scope Hierarchy
Table 37:	Error Code Types
Table 38:	Error code: transaction type (TT)
Table 39:	Error codes: cache level (LL)
Table 40:	Error codes: memory transaction type (RRRR)
Table 41:	Blocks Capable of Supporting MCA Banks
Table 42:	Mapping of Blocks to MCA_IPID[HwId] and MCA_IPID[McaType]
Table 43:	Legacy MCA Registers
Table 44:	MCAX Registers
Table 45:	Core MCA Bank to Block Mapping
Table 46:	Non-core MCA Bank to Block Mapping
Table 47:	Non-core MCA Bank to Block Mapping for SDP1 products
Table 48:	Non-core MCA Bank to Block Mapping for SDP-wide products
Table 49:	Non-core MCA Bank to Block Mapping for 64-bank products
Table 50:	Non-core MCA Bank to Block Mapping for 64-bank SDP1 products
Table 51:	Non-core MCA Bank to Block Mapping for 64-bank SDP-wide products
Table 52:	MCA_STATUS_LS
Table 53:	MCA_ADDR_LS
Table 54:	MCA_SYND_LS
Table 55:	MCA_STATUS_IF
Table 56:	MCA_ADDR_IF
Table 57:	MCA_SYND_IF
Table 58:	MCA_STATUS_L2
Table 59:	MCA_ADDR_L2
Table 60:	MCA_SYND_L2
Table 61:	MCA_STATUS_DE
Table 62:	MCA_ADDR_DE
Table 63:	MCA_SYND_DE
Table 64:	MCA_STATUS_EX
Table 65:	MCA_ADDR_EX
Table 66:	MCA_SYND_EX
Table 67:	MCA_STATUS_FP
Table 68:	MCA_ADDR_FP
Table 69:	MCA_SYND_FP
Table 70:	MCA_STATUS_L3
Table 71:	MCA_ADDR_L3
Table 72:	MCA_SYND_L3
Table 73:	MCA_STATUS_CS
Table 74:	MCA_ADDR_CS
Table 75:	MCA_SYND_CS
Table 76:	MCA_STATUS_PIE
Table 77:	MCA_ADDR_PIE
Table 78:	MCA_SYND_PIE
Table 79:	MCA_STATUS_UMC
Table 80:	MCA_ADDR_UMC

Table 81:	MCA_SYND_UMC
Table 82:	MCA_STATUS_PSP
Table 83:	MCA_ADDR_PSP
Table 84:	MCA_SYND_PSP
Table 85:	MCA_STATUS_SMU
Table 86:	MCA_ADDR_SMU
Table 87:	MCA_SYND_SMU
Table 88:	MCA_STATUS_MP5
Table 89:	MCA_ADDR_MP5
Table 90:	MCA_SYND_MP5
Table 91:	MCA_STATUS_NBIO
Table 92:	MCA_ADDR_NBIO
Table 93:	MCA_SYND_NBIO
Table 94:	MCA_STATUS_PCIE
Table 95:	MCA_ADDR_PCIE
Table 96:	MCA_SYND_PCIE
Table 97:	MCA_STATUS_KPX_SERDES
Table 98:	MCA_ADDR_KPX_SERDES
Table 99:	MCA_SYND_KPX_SERDES
Table 100:	MCA_STATUS_KPX_GMI
Table 101:	MCA_ADDR_KPX_GMI
Table 102:	MCA_SYND_KPX_GMI
Table 103:	MCA_STATUS_PCS_GMI
Table 104:	MCA_ADDR_PCS_GMI
Table 105:	MCA_SYND_PCS_GMI
Table 106:	MCA_STATUS_PCS_XGMI
Table 107:	MCA_ADDR_PCS_XGMI
Table 108:	MCA_SYND_PCS_XGMI
Table 109:	MCA_STATUS_NBIF
Table 110:	MCA_ADDR_NBIF
Table 111:	MCA_SYND_NBIF
Table 112:	MCA_STATUS_SHUB
Table 113:	MCA_ADDR_SHUB
Table 114:	MCA_SYND_SHUB
Table 115:	MCA_STATUS_USB
Table 116:	MCA_ADDR_USB
Table 117:	MCA_SYND_USB
Table 118:	MCA_STATUS_SATA
Table 119:	MCA_ADDR_SATA
Table 120:	MCA_SYND_SATA
Table 121:	MCA_STATUS_MPDMA
Table 122:	MCA_ADDR_MPDMA
Table 123:	MCA_SYND_MPDMA
Table 124:	AER Severity for Root Complex and Switch Downstream Ports
Table 125:	AER Severity for Endpoint Devices and Switch Upstream Ports

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 31: Machine Check Terms and Acronyms

Term	Description
MCA	Machine Check Architecture.
MCAX	Machine Check Architecture eXtensions.
WRIG	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 32 [Legacy MCA MSR Layout].

Table 32: 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_DESTSTATUS
 - 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 33 [MCAX MSR Layout].

Access to unused MCAX MSRs is RAZ/WRIG. MCA Bank 4 is always Read-as-zero (RAZ/WRIG).

Table 33: 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 34 [MCAX Implementation-Specific Register Layout].

Table 34: 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 35 [Error Overwrite Priorities] indicates which errors are overwritten in the error status registers.

Table 35: Error Overwrite Priorities

		Older Error		
		Uncorrected	Deferred	Corrected
Newer	Uncorrected	-	Overwrite	Overwrite
	Deferred	-	-	Overwrite

Error	Corrected	-	-	-
-------	-----------	---	---	---

Table 36 [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 36: 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 37: 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 38: Error code: transaction type (TT)

TT	Transaction Type
00	Instruction
01	Data
10	Generic
11	Reserved

Table 39: Error codes: cache level (LL)

LL	Cache Level
00	L0: Core
01	L1: Level 1
10	L2: Level 2
11	LG: Generic

Table 40: 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 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::Msrr::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::Msrr::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[Mcax]). An MCA bank is an MCAX bank if MCA_CONFIG[Mcax] == 1 in that bank.

3.1.5.2 Communicating MCAX Support

Software which supports MCAX must set MCA_CONFIG[McaxEn] = 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::Msrr::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::Msrr::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[McaEn] = 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::Msrb::[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 41: 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
PB	Parameter Block
PSP	Platform Security Processor
SMU	System Management Controller Unit
PCIE	PCIe® Root Port
MP5	Microprocessor5 Management Controller
NBIF	On-chip PCIe Bus Interface
SHUB	System Hub
USB	USB Controller
SATA	SATA Controller
PCS_GMI	GMI Controller
PCS_XGMI	XGMI Controller
KPX_SERDES	High Speed Interface Unit
MPIO	IO Microprocessor
KPX_GMI	High Speed Interface Unit (GMI)
MPDMA	DMA Engine Controller

Table 42: 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
MP5	0x1	0x2
PCS_GMI	0x241	0x0
KPX_GMI	0x269	0x0
MPDMA	0x1	0x3
UMC	0x96	0x0
PCIE	0x46	0x1
SATA	0xa8	0x0
USB	0xaa	0x0
NBIO	0x18	0x0
NBIF	0x6c	0x0
SMU	0x1	0x1
SHUB	0x80	0x0
PIE	0x2e	0x1
PSP	0xff	0x1
PCS_XGMI	0x50	0x0
KPX_SERDES	0x259	0x0
CS	0x2e	0x2
EX	0xb0	0x5
FP	0xb0	0x6
DE	0xb0	0x3

3.2.2 Implemented Machine Check Bank Registers

Table 43 [Legacy MCA Registers] provides links to the description of each block's Legacy MCA registers. Table 44 [MCAX Registers] provides links to the description of each block's MCA Extension Registers.

Table 43: 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
PSP	MCA::PSP::MCA_CTL_PSP	MCA::PSP::MCA_STATUS_PSP	MCA::PSP::MCA_ADDR_PSP	MCA::PSP::MCA_MISC0_PSP	MCA::PSP::MCA_CTL_MASK_PSP
MP5	MCA::MP5::MCA_CTL_MP5	MCA::MP5::MCA_STATUS_MP5	MCA::MP5::MCA_ADDR_MP5	MCA::MP5::MCA_MISC0_MP5	MCA::MP5::MCA_CTL_MASK_MP5
SMU	MCA::SMU::MCA_CTL_SMU_MP1 MCA::SMU::MCA_CTL_SMU_MPIO MCA::SMU::MCA_CTL_SMU_MPRAS	MCA::SMU::MCA_STATUS_SMU	MCA::SMU::MCA_ADDR_SMU	MCA::SMU::MCA_MISC0_SMU	MCA::SMU::MCA_CTL_MASK_SMU_MP1 MCA::SMU::MCA_CTL_MASK_SMU_MPIO MCA::SMU::MCA_CTL_MASK_SMU_MPRAS
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
PCIE	MCA::PCIE::MCA_CTL_PCIE	MCA::PCIE::MCA_STATUS_PCIE	MCA::PCIE::MCA_ADDR_PCIE	MCA::PCIE::MCA_MISC0_PCIE	MCA::PCIE::MCA_CTL_MASK_PCIE
NBIF	MCA::NBIF::MCA_CTL_NBIF	MCA::NBIF::MCA_STATUS_NBIF	MCA::NBIF::MCA_ADDR_NBIF	MCA::NBIF::MCA_MISC0_NBIF	MCA::NBIF::MCA_CTL_MASK_NBIF
SHUB	MCA::SHUB::MCA_CTL_SHUB	MCA::SHUB::MCA_STATUS_SHUB	MCA::SHUB::MCA_ADDR_SHUB	MCA::SHUB::MCA_MISC0_SHUB	MCA::SHUB::MCA_CTL_MASK_SHUB
KPX_SERDES	MCA::KPX::SERDES::MCA_CTL_KPX_SERDES	MCA::KPX::SERDES::MCA_STATUS_KPX_SERDES	MCA::KPX::SERDES::MCA_ADDR_KPX_SERDES	MCA::KPX::SERDES::MCA_MISC0_KPX_SERDES	MCA::KPX::SERDES::MCA_CTL_MASK_KPX_SERDES
KPX_GMI	MCA::KPX::GMI::MCA_CTL_KPX_GMI	MCA::KPX::GMI::MCA_STATUS_KPX_GMI	MCA::KPX::GMI::MCA_ADDR_KPX_GMI	MCA::KPX::GMI::MCA_MISC0_KPX_GMI	MCA::KPX::GMI::MCA_CTL_MASK_KPX_GMI
PCS_GMI	MCA::PCS::GMI::MCA_CTL_PCS_GMI	MCA::PCS::GMI::MCA_STATUS_PCS_GMI	MCA::PCS::GMI::MCA_ADDR_PCS_GMI	MCA::PCS::GMI::MCA_MISC0_PCS_GMI	MCA::PCS::GMI::MCA_CTL_MASK_PCS_GMI
PCS_XGMI	MCA::PCS::XGMI::MCA_CTL_PCS_XGMI	MCA::PCS::XGMI::MCA_STATUS_PCS_XGMI	MCA::PCS::XGMI::MCA_ADDR_PCS_XGMI	MCA::PCS::XGMI::MCA_MISC0_PCS_XGMI	MCA::PCS::XGMI::MCA_CTL_MASK_PCS_XGMI
MPDMA	MCA::MPDMA::MCA_CTL_MPDMA	MCA::MPDMA::MCA_STATUS_MPDMA	MCA::MPDMA::MCA_ADDR_MPDMA	MCA::MPDMA::MCA_MISC0_MPDMA	MCA::MPDMA::MCA_CTL_MASK_MPDMA
SATA	MCA::SATA::MCA_CTL_SATA	MCA::SATA::MCA_STATUS_SATA	MCA::SATA::MCA_ADDR_SATA	MCA::SATA::MCA_MISC0_SATA	MCA::SATA::MCA_CTL_MASK_SATA
USB	MCA::USB::MCA_CTL_USB	MCA::USB::MCA_STATUS_USB	MCA::USB::MCA_ADDR_USB	MCA::USB::MCA_MISC0_USB	MCA::USB::MCA_CTL_MASK_USB

Table 44: MCA 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_SYND_1_UMC MCA::UMC::MCA_SYND_2_UMC	MCA::UMC::MCA_DESTAT_UMC	MCA::UMC::MCA_DEADDR_UMC

PSP	MCA::PSP::MCA_CONFIG_PSP	MCA::PSP::MCA_IPID_PSP	MCA::PSP::MCA_SYND_PSP	--	--
SMU	MCA::SMU::MCA_CONFIG_SMU	MCA::SMU::MCA_IPID_SMU	MCA::SMU::MCA_SYND_SMU	--	--
MP5	MCA::MP5::MCA_CONFIG_MP5	MCA::MP5::MCA_IPID_MP5	MCA::MP5::MCA_SYND_MP5	--	--
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
PCIE	MCA::PCIE::MCA_CONFIG_PCIE	MCA::PCIE::MCA_IPID_PCIE	MCA::PCIE::MCA_SYND_PCIE	MCA::PCIE::MCA_DESTAT_PCIE	MCA::PCIE::MCA_DEADDR_PCIE
NBIF	MCA::NBIF::MCA_CONFIG_NBIF	MCA::NBIF::MCA_IPID_NBIF	MCA::NBIF::MCA_SYND_NBIF	MCA::NBIF::MCA_DESTAT_NBIF	MCA::NBIF::MCA_DEADDR_NBIF
SHUB	MCA::SHUB::MCA_CONFIG_SHUB	MCA::SHUB::MCA_IPID_SHUB	MCA::SHUB::MCA_SYND_SHUB	MCA::SHUB::MCA_DESTAT_SHUB	MCA::SHUB::MCA_DEADDR_SHUB
KPX_SERDES	MCA::KPX::SERDES::MCA_CONFIG_KPX_SERDES	MCA::KPX::SERDES::MCA_IPID_KPX_SERDES	MCA::KPX::SERDES::MCA_SYND_KPX_SERDES	--	--
KPX_GMI	MCA::KPX::GMI::MCA_CONFIG_KPX_GMI	MCA::KPX::GMI::MCA_IPID_KPX_GMI	MCA::KPX::GMI::MCA_SYND_KPX_GMI	--	--
PCS_GMI	MCA::PCS::GMI::MCA_CONFIG_PCS_GMI	MCA::PCS::GMI::MCA_IPID_PCS_GMI	MCA::PCS::GMI::MCA_SYND_PCS_GMI	--	--
PCS_XGMI	MCA::PCS::XGMI::MCA_CONFIG_PCS_XGMI	MCA::PCS::XGMI::MCA_IPID_PCS_XGMI	MCA::PCS::XGMI::MCA_SYND_PCS_XGMI	--	--
MPDMA	MCA::MPDMA::MCA_CONFIG_MPDMA	MCA::MPDMA::MCA_IPID_MPDMA	MCA::MPDMA::MCA_SYND_MPDMA	--	--
SATA	MCA::SATA::MCA_CONFIG_SATA	MCA::SATA::MCA_IPID_SATA	MCA::SATA::MCA_SYND_SATA	--	--
USB	MCA::USB::MCA_CONFIG_USB	MCA::USB::MCA_IPID_USB	MCA::USB::MCA_SYND_USB	--	--

3.2.3 Mapping of Banks to Blocks

Table 45 [Core MCA Bank to Block Mapping] shows MCA banks that are present in the address space of every logical core.

Table 45: Core MCA Bank to Block Mapping

Bank	Block
0	LS
1	IF
2	L2
3	DE
4	RAZ
5	EX
6	FP

Table 46 [Non-core MCA Bank to Block Mapping] shows MCA banks that are present in the address space of specific logical cores.

Table 46: Non-core MCA Bank to Block Mapping

Bank	Thread 0	Thread 2	Thread 4	Thread 6	Thread 8	Thread 10	Thread 12	Thread 14	Thread 16	Thread 18	Thread 20	Thread 22	Thread 24	Thread 26	Thread 28	Thread 30	Thread 32	Thread 34	Thread 36	Thread 38	Thread 40	Thread 42	Thread 44	Thread 46
0	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS
1	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF
2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2

[illegible]

54	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
55	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
56	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
57	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
58	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
59	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
60	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
61	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
62	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
63	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ

Table 47 [Non-core MCA Bank to Block Mapping for SDP1 products]

Table 47: Non-core MCA Bank to Block Mapping for SDP1 products

Bank	Thread 0	Thread 2	Thread 4	Thread 6	Thread 8	Thread 10	Thread 12	Thread 14	Thread 16	Thread 18	Thread 20	Thread 22	Thread 24	Thread 26	Thread 28	Thread 30	Thread 32	Thread 34	Thread 36	Thread 38	Thread 40	Thread 42	Thread 44	Thread 46
0	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS
1	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF
2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2
3	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE
4	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
5	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX
6	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP
7	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3
8	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3
9	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3
10	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3
11	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3
12	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3
13	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3
14	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3	L3
15	MP5	RAZ	MP5	RAZ	MP5	RAZ	MP5	RAZ	MP5	RAZ	MP5	RAZ	MP5	RAZ	MP5	RAZ	MP5	RAZ	MP5	RAZ	MP5	RAZ	MP5	RAZ
16	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ
17	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI	RAZ	PCS_GMI
18	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ
19	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI	RAZ	KPX_GMI
20	MPDMA	MPDMA	MPDMA	MPDMA	MPDMA	MPDMA	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
21	UMC	UMC	UMC	UMC	UMC	UMC	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
22	UMC	UMC	UMC	UMC	UMC	UMC	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
23	CS	CS	CS	CS	CS	CS	CS	CS	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
24	CS	CS	CS	CS	CS	CS	CS	CS	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
25	PCIE	PCIE	PCIE	PCIE	PCIE	PCIE	PCIE	PCIE	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
26	SATA	SATA	SATA	SATA	USB	USB	RAZ	PCIE	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
27	NBIO	NBIO	NBIO	NBIO	NBIO	NBIO	NBIO	NBIO	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
28	NBIF	NBIF	NBIF	NBIF	NBIF	NBIF	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
29	SMU	SMU	SMU	SHUB	SHUB	SHUB	SHUB	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
30	PIE	PSP	PCS_XGMI	PCS_XGMI	PCS_XGMI	PCS_XGMI	PCS_XGMI	PCS_XGMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
31	KPX_SER	KPX_SER	KPX_SER	KPX_SER	KPX_SER	KPX_SER	KPX_SER	KPX_SER	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ

[illegible]

Table 48 [Non-core MCA Bank to Block Mapping for SDP-wide products]

Table 48: Non-core MCA Bank to Block Mapping for SDP-wide products

[illegible]

18	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI
19	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI
20	MPDM_A	MPDM_A	MPDM_A	MPDM_A	MPDM_A	MPDM_A	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
21	UMC	UMC	UMC	UMC	UMC	UMC	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
22	UMC	UMC	UMC	UMC	UMC	UMC	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
23	CS	CS	CS	CS	CS	CS	CS	CS	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
24	CS	CS	CS	CS	CS	CS	CS	CS	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
25	PCIE	PCIE	PCIE	PCIE	PCIE	PCIE	PCIE	PCIE	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
26	SATA	SATA	SATA	SATA	USB	USB	RAZ	PCIE	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
27	NBIO	NBIO	NBIO	NBIO	NBIO	NBIO	NBIO	NBIO	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
28	NBIF	NBIF	NBIF	NBIF	NBIF	NBIF	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
29	SMU	SMU	SMU	SHUB	SHUB	SHUB	SHUB	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
30	PIE	PSP	PCS_X_GMI	PCS_X_GMI	PCS_X_GMI	PCS_X_GMI	PCS_X_GMI	PCS_X_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
31	KPX_S_ERDES	KPX_S_ERDES	KPX_S_ERDES	KPX_S_ERDES	KPX_S_ERDES	KPX_S_ERDES	KPX_S_ERDES	KPX_S_ERDES	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
32	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
33	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
34	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
35	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
36	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
37	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
38	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
39	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
40	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
41	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
42	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
43	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
44	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
45	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
46	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
47	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
48	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
49	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
50	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
51	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
52	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
53	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
54	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
55	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
56	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
57	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
58	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
59	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
60	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
61	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
62	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
63	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ

Table 49 [Non-core MCA Bank to Block Mapping for 64-bank products]

Table 49: Non-core MCA Bank to Block Mapping for 64-bank products

Bank	Thread 0	Thread 2	Thread 4	Thread 6	Thread 8	Thread 10	Thread 12	Thread 14	Thread 16	Thread 18	Thread 20	Thread 22
0	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS
1	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF
2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2

3	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE
4	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
5	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX
6	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP
7	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
8	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
9	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
10	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
11	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
12	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
13	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
14	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
15	MP5	MP5	MP5	MP5	MP5	MP5	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
16	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
17	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
18	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
19	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
20	MPDMA	MPDMA	MPDMA	MPDMA	MPDMA	MPDMA	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
21	UMC	UMC	UMC	UMC	UMC	UMC	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
22	UMC	UMC	UMC	UMC	UMC	UMC	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
23	CS	CS	CS	CS	CS	CS	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
24	CS	CS	CS	CS	CS	CS	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
25	PCIE	PCIE	PCIE	PCIE	PCIE	PCIE	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
26	SATA	SATA	SATA	SATA	USB	USB	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
27	NBIO	NBIO	NBIO	NBIO	NBIO	NBIO	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
28	NBIF	NBIF	NBIF	NBIF	NBIF	NBIF	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
29	SMU	SMU	SMU	SHUB	SHUB	SHUB	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
30	PIE	PSP	PCS_XG MI	PCS_XG MI	PCS_XG MI	PCS_XG MI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
31	KPX_SER DES	KPX_SER DES	KPX_SER DES	KPX_SER DES	KPX_SER DES	KPX_SER DES	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
32	CS	CS	CS	CS	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
33	PCIE	PCIE	PCIE	RAZ	RAZ	SHUB	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
34	NBIO	NBIO	PCS_XG MI	PCS_XG MI	KPX_SER DES	KPX_SER DES	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
35	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
36	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
37	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
38	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
39	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
40	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
41	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
42	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
43	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
44	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
45	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
46	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
47	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
48	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
49	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
50	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
51	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
52	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
53	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
54	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
55	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
56	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
57	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
58	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
59	MP5	MP5	MP5	MP5	MP5	MP5	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ

60	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
61	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
62	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
63	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ

Table 50 [Non-core MCA Bank to Block Mapping for 64-bank SDP1 products]

Table 50: Non-core MCA Bank to Block Mapping for 64-bank SDP1 products

Bank	Thread 0	Thread 2	Thread 4	Thread 6	Thread 8	Thread 10	Thread 12	Thread 14	Thread 16	Thread 18	Thread 20	Thread 22
0	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS	LS
1	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF	IF
2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2	L2
3	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE	DE
4	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
5	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX	EX
6	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP	FP
7	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
8	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
9	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
10	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
11	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
12	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
13	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
14	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
15	MP5	MP5	MP5	MP5	MP5	MP5	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
16	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
17	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
18	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
19	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
20	MPDMA	MPDMA	MPDMA	MPDMA	MPDMA	MPDMA	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
21	UMC	UMC	UMC	UMC	UMC	UMC	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
22	UMC	UMC	UMC	UMC	UMC	UMC	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
23	CS	CS	CS	CS	CS	CS	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
24	CS	CS	CS	CS	CS	CS	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
25	PCIE	PCIE	PCIE	PCIE	PCIE	PCIE	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
26	SATA	SATA	SATA	SATA	USB	USB	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
27	NBIO	NBIO	NBIO	NBIO	NBIO	NBIO	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
28	NBIF	NBIF	NBIF	NBIF	NBIF	NBIF	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
29	SMU	SMU	SMU	SHUB	SHUB	SHUB	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
30	PIE	PSP	PCS_XG MI	PCS_XG MI	PCS_XG MI	PCS_XG MI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
31	KPX_SER DES	KPX_SER DES	KPX_SER DES	KPX_SER DES	KPX_SER DES	KPX_SER DES	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
32	CS	CS	CS	CS	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
33	PCIE	PCIE	PCIE	RAZ	RAZ	SHUB	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
34	NBIO	NBIO	PCS_XG MI	PCS_XG MI	KPX_SER DES	KPX_SER DES	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
35	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
36	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
37	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
38	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
39	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
40	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
41	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
42	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
43	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
44	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
45	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ

46	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
47	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
48	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
49	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
50	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
51	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
52	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
53	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
54	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
55	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
56	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
57	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
58	L3	L3	L3	L3	L3	L3	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
59	MP5	MP5	MP5	MP5	MP5	MP5	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
60	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
61	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
62	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
63	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ

Table 51 [Non-core MCA Bank to Block Mapping for 64-bank SDP-wide products]

Table 51: Non-core MCA Bank to Block Mapping for 64-bank SDP-wide products

Bank	Thread 0	Thread 2	Thread 4	Thread 6	Thread 8	Thread 10	Thread 12	Thread 14
0	LS	LS	LS	LS	LS	LS	LS	LS
1	IF	IF	IF	IF	IF	IF	IF	IF
2	L2	L2	L2	L2	L2	L2	L2	L2
3	DE	DE	DE	DE	DE	DE	DE	DE
4	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
5	EX	EX	EX	EX	EX	EX	EX	EX
6	FP	FP	FP	FP	FP	FP	FP	FP
7	L3	L3	L3	L3	L3	L3	RAZ	RAZ
8	L3	L3	L3	L3	L3	L3	RAZ	RAZ
9	L3	L3	L3	L3	L3	L3	RAZ	RAZ
10	L3	L3	L3	L3	L3	L3	RAZ	RAZ
11	L3	L3	L3	L3	L3	L3	RAZ	RAZ
12	L3	L3	L3	L3	L3	L3	RAZ	RAZ
13	L3	L3	L3	L3	L3	L3	RAZ	RAZ
14	L3	L3	L3	L3	L3	L3	RAZ	RAZ
15	MP5	MP5	MP5	MP5	MP5	MP5	RAZ	RAZ
16	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	RAZ	RAZ
17	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	RAZ	RAZ
18	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	RAZ	RAZ
19	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	RAZ	RAZ
20	MPDMA	MPDMA	MPDMA	MPDMA	MPDMA	MPDMA	RAZ	RAZ
21	UMC	UMC	UMC	UMC	UMC	UMC	RAZ	RAZ
22	UMC	UMC	UMC	UMC	UMC	UMC	RAZ	RAZ
23	CS	CS	CS	CS	CS	CS	RAZ	RAZ
24	CS	CS	CS	CS	CS	CS	RAZ	RAZ
25	PCIE	PCIE	PCIE	PCIE	PCIE	PCIE	RAZ	RAZ
26	SATA	SATA	SATA	SATA	USB	USB	RAZ	RAZ
27	NBIO	NBIO	NBIO	NBIO	NBIO	NBIO	RAZ	RAZ

28	NBIF	NBIF	NBIF	NBIF	NBIF	NBIF	RAZ	RAZ
29	SMU	SMU	SMU	SHUB	SHUB	SHUB	RAZ	RAZ
30	PIE	PSP	PCS_XGM I	PCS_XGM I	PCS_XGM I	PCS_XGM I	RAZ	RAZ
31	KPX_SER DES	KPX_SER DES	KPX_SER DES	KPX_SER DES	KPX_SER DES	KPX_SER DES	RAZ	RAZ
32	CS	CS	CS	CS	RAZ	RAZ	RAZ	RAZ
33	PCIE	PCIE	PCIE	RAZ	RAZ	SHUB	RAZ	RAZ
34	NBIO	NBIO	PCS_XGM I	PCS_XGM I	KPX_SER DES	KPX_SER DES	RAZ	RAZ
35	L3	L3	L3	L3	L3	L3	RAZ	RAZ
36	L3	L3	L3	L3	L3	L3	RAZ	RAZ
37	L3	L3	L3	L3	L3	L3	RAZ	RAZ
38	L3	L3	L3	L3	L3	L3	RAZ	RAZ
39	L3	L3	L3	L3	L3	L3	RAZ	RAZ
40	L3	L3	L3	L3	L3	L3	RAZ	RAZ
41	L3	L3	L3	L3	L3	L3	RAZ	RAZ
42	L3	L3	L3	L3	L3	L3	RAZ	RAZ
43	L3	L3	L3	L3	L3	L3	RAZ	RAZ
44	L3	L3	L3	L3	L3	L3	RAZ	RAZ
45	L3	L3	L3	L3	L3	L3	RAZ	RAZ
46	L3	L3	L3	L3	L3	L3	RAZ	RAZ
47	L3	L3	L3	L3	L3	L3	RAZ	RAZ
48	L3	L3	L3	L3	L3	L3	RAZ	RAZ
49	L3	L3	L3	L3	L3	L3	RAZ	RAZ
50	L3	L3	L3	L3	L3	L3	RAZ	RAZ
51	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
52	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
53	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
54	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
55	PCS_GMI	PCS_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
56	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	RAZ	RAZ
57	KPX_GMI	KPX_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
58	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	RAZ	RAZ
59	MP5	MP5	MP5	MP5	MP5	MP5	RAZ	RAZ
60	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	PCS_GMI	RAZ	RAZ
61	PCS_GMI	PCS_GMI	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ
62	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	KPX_GMI	RAZ	RAZ
63	KPX_GMI	KPX_GMI	RAZ	RAZ	RAZ	RAZ	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 41 [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.	
_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst0_n[383:0]_aliasMSRLEGACY; MSR0000_0400	
_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst0_n[383:0]_aliasMSR; MSRC000_2000	
Bits	Description
63:27	Reserved.
26	Hwa. Read-write. Reset: 0. Hardware Asserts (HWAs)
25	SystemReadDataErrorWcb. 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	ScbDataNonCacheable. Read-write. Reset: 0. Error on SCB data for non-cacheable DRAM or IO, discovered at data-pull time
23	ScbData1. Read-write. Reset: 0. Error on SCB data, commit pipe 1, discovered at SCB commit time
22	ScbData0. Read-write. Reset: 0. Error on SCB data, commit pipe 0, discovered at SCB commit time
21	ScbStateAddr. Read-write. Reset: 0. Error on SCB cacheline state (way and moesi state) or address field
20	L2DataErr. Read-write. Reset: 0. L2 Fill Data error.
19	DcTagErr7. Read-write. Reset: 0. DC Tag error type 5.
18	DcTagErr3. Read-write. Reset: 0. DC Tag error type 3.
17	PDC. Read-write. Reset: 0. PDC parity error.
16	L2DTLB. Read-write. Reset: 0. Level 2 TLB parity error.
15	DcTagErr4. Read-write. Reset: 0. DC Tag error type 4.
14	DcDataErr3. Read-write. Reset: 0. DC Data error type 3.
13	DcDataErr2. Read-write. Reset: 0. DC Data error type 2.
12	DcDataErr1. Read-write. Reset: 0. DC Data error type 1 and poison consumption. MCA_STATUS[Poison] is set on poison consumption from L2/L3.
11	DcTagErr2. Read-write. Reset: 0. DC Tag error type 2.
10	SystemReadDataErrorMab. 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	SystemReadDataErrorUcode. 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	IntErrTyp2. Read-write. Reset: 0. Internal error type 2.
7	IntErrTyp1. Read-write. Reset: 0. Internal error type 1.
6	DcTagErr1. Read-write. Reset: 0. DC Tag error type 1.
5	DcTagErr6. Read-write. Reset: 0. DC Tag error type 6.
4	DcTagErr5. Read-write. Reset: 0. DC Tag error type 5.
3	L1DTLB. Read-write. Reset: 0. Level 1 TLB parity error.
2	MAB. Read-write. Reset: 0. Miss address buffer payload parity error.
1	STQ. Read-write. Reset: 0. Store queue parity error.
0	LDQ. 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst0_n[383:0]_aliasMSRSLLEGACY; MSR0000_0001

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst0_n[383:0]_aliasMSRLEGACY; MSR0000_0401

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst0_n[383:0]_aliasMSR; MSRC000_2001

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. 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	MiscV. 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	AddrV. 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	PCC. 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	ErrCoreIdVal. 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	TCC. 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	RESERV54. 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	SyndV. 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	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. 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	RESERV22. 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	ErrorCodeExt. 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	ErrorCode. 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_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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst0_n[383:0]_aliasMSRSLLEGACY; MSR0000_0000

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst0_n[383:0]_aliasMSRLEGACY; MSR0000_0402

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst0_n[383:0]_aliasMSR; MSRC000_2002

Bits	Description
63:0	ErrorAddr. 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 53: 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst0_n[383:0]_aliasMSRLEGACY; MSR0000_0403

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst0_n[383:0]_aliasMSR; MSRC000_2003

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr:: HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr:: HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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	LvtOffset. 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	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr:: HWCR[McStatusWrEn] !MCA::LS::MCA_MISC0_LS[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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::LS::MCA_MISC0_LS[Locked]) ? Read-write : Read-only.
48	Ovrflw. 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	ErrCnt. 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	BlkPtr. 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst0_n[383:0]_aliasMSR; MSRC000_2004

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	LogDeferredInMcaStat. 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	McaXEnable. 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	IntPresent. Read-only. Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::LS::MCA_CONFIG_LS[IntEn].
9	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,1. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::LS::MCA_CONFIG_LS[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. 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	DeferredErrorLoggingSupported. 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	McaX. 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst0_n[383:0]_aliasMSR; MSRC000_2005

Bits	Description
63:48	McaType . Read-only. Reset: 0000h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi . 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	HardwareID . Read-only. Reset: 0B0h. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _ccd0_lthree0_core0_thread0_inst0_n0_aliasMSR: 2000_A200h
	Init: _ccd0_lthree0_core0_thread1_inst0_n1_aliasMSR: 2000_A300h
	Init: _ccd0_lthree0_core1_thread0_inst0_n2_aliasMSR: 2002_A200h
	Init: _ccd0_lthree0_core1_thread1_inst0_n3_aliasMSR: 2002_A300h
	Init: _ccd0_lthree0_core2_thread0_inst0_n4_aliasMSR: 2004_A200h
	Init: _ccd0_lthree0_core2_thread1_inst0_n5_aliasMSR: 2004_A300h
	Init: _ccd0_lthree0_core3_thread0_inst0_n6_aliasMSR: 2006_A200h
	Init: _ccd0_lthree0_core3_thread1_inst0_n7_aliasMSR: 2006_A300h
	Init: _ccd0_lthree0_core4_thread0_inst0_n8_aliasMSR: 2008_A200h
	Init: _ccd0_lthree0_core4_thread1_inst0_n9_aliasMSR: 2008_A300h
	Init: _ccd0_lthree0_core5_thread0_inst0_n10_aliasMSR: 200A_A200h
	Init: _ccd0_lthree0_core5_thread1_inst0_n11_aliasMSR: 200A_A300h
	Init: _ccd0_lthree0_core6_thread0_inst0_n12_aliasMSR: 200C_A200h
	Init: _ccd0_lthree0_core6_thread1_inst0_n13_aliasMSR: 200C_A300h
	Init: _ccd0_lthree0_core7_thread0_inst0_n14_aliasMSR: 200E_A200h
	Init: _ccd0_lthree0_core7_thread1_inst0_n15_aliasMSR: 200E_A300h
	Init: _ccd0_lthree0_core8_thread0_inst0_n16_aliasMSR: 2010_A200h
	Init: _ccd0_lthree0_core8_thread1_inst0_n17_aliasMSR: 2010_A300h
	Init: _ccd0_lthree0_core9_thread0_inst0_n18_aliasMSR: 2012_A200h
	Init: _ccd0_lthree0_core9_thread1_inst0_n19_aliasMSR: 2012_A300h
	Init: _ccd0_lthree0_core10_thread0_inst0_n20_aliasMSR: 2014_A200h
	Init: _ccd0_lthree0_core10_thread1_inst0_n21_aliasMSR: 2014_A300h
	Init: _ccd0_lthree0_core11_thread0_inst0_n22_aliasMSR: 2016_A200h
	Init: _ccd0_lthree0_core11_thread1_inst0_n23_aliasMSR: 2016_A300h
	Init: _ccd0_lthree0_core12_thread0_inst0_n24_aliasMSR: 2018_A200h
	Init: _ccd0_lthree0_core12_thread1_inst0_n25_aliasMSR: 2018_A300h
	Init: _ccd0_lthree0_core13_thread0_inst0_n26_aliasMSR: 201A_A200h
	Init: _ccd0_lthree0_core13_thread1_inst0_n27_aliasMSR: 201A_A300h
	Init: _ccd0_lthree0_core14_thread0_inst0_n28_aliasMSR: 201C_A200h
	Init: _ccd0_lthree0_core14_thread1_inst0_n29_aliasMSR: 201C_A300h
	Init: _ccd0_lthree0_core15_thread0_inst0_n30_aliasMSR: 201E_A200h
	Init: _ccd0_lthree0_core15_thread1_inst0_n31_aliasMSR: 201E_A300h
	Init: _ccd1_lthree0_core0_thread0_inst0_n32_aliasMSR: 2080_A200h
	Init: _ccd1_lthree0_core0_thread1_inst0_n33_aliasMSR: 2080_A300h
	Init: _ccd1_lthree0_core1_thread0_inst0_n34_aliasMSR: 2082_A200h
	Init: _ccd1_lthree0_core1_thread1_inst0_n35_aliasMSR: 2082_A300h
	Init: _ccd1_lthree0_core2_thread0_inst0_n36_aliasMSR: 2084_A200h
	Init: _ccd1_lthree0_core2_thread1_inst0_n37_aliasMSR: 2084_A300h
	Init: _ccd1_lthree0_core3_thread0_inst0_n38_aliasMSR: 2086_A200h
	Init: _ccd1_lthree0_core3_thread1_inst0_n39_aliasMSR: 2086_A300h
	Init: _ccd1_lthree0_core4_thread0_inst0_n40_aliasMSR: 2088_A200h
	Init: _ccd1_lthree0_core4_thread1_inst0_n41_aliasMSR: 2088_A300h
	Init: _ccd1_lthree0_core5_thread0_inst0_n42_aliasMSR: 208A_A200h
	Init: _ccd1_lthree0_core5_thread1_inst0_n43_aliasMSR: 208A_A300h
	Init: _ccd1_lthree0_core6_thread0_inst0_n44_aliasMSR: 208C_A200h
	Init: _ccd1_lthree0_core6_thread1_inst0_n45_aliasMSR: 208C_A300h
	Init: _ccd1_lthree0_core7_thread0_inst0_n46_aliasMSR: 208E_A200h
	Init: _ccd1_lthree0_core7_thread1_inst0_n47_aliasMSR: 208E_A300h
	Init: _ccd1_lthree0_core8_thread0_inst0_n48_aliasMSR: 2090_A200h

Init: _ccd1_lthree0_core8_thread1_inst0_n49_aliasMSR: 2090_A300h
Init: _ccd1_lthree0_core9_thread0_inst0_n50_aliasMSR: 2092_A200h
Init: _ccd1_lthree0_core9_thread1_inst0_n51_aliasMSR: 2092_A300h
Init: _ccd1_lthree0_core10_thread0_inst0_n52_aliasMSR: 2094_A200h
Init: _ccd1_lthree0_core10_thread1_inst0_n53_aliasMSR: 2094_A300h
Init: _ccd1_lthree0_core11_thread0_inst0_n54_aliasMSR: 2096_A200h
Init: _ccd1_lthree0_core11_thread1_inst0_n55_aliasMSR: 2096_A300h
Init: _ccd1_lthree0_core12_thread0_inst0_n56_aliasMSR: 2098_A200h
Init: _ccd1_lthree0_core12_thread1_inst0_n57_aliasMSR: 2098_A300h
Init: _ccd1_lthree0_core13_thread0_inst0_n58_aliasMSR: 209A_A200h
Init: _ccd1_lthree0_core13_thread1_inst0_n59_aliasMSR: 209A_A300h
Init: _ccd1_lthree0_core14_thread0_inst0_n60_aliasMSR: 209C_A200h
Init: _ccd1_lthree0_core14_thread1_inst0_n61_aliasMSR: 209C_A300h
Init: _ccd1_lthree0_core15_thread0_inst0_n62_aliasMSR: 209E_A200h
Init: _ccd1_lthree0_core15_thread1_inst0_n63_aliasMSR: 209E_A300h
Init: _ccd2_lthree0_core0_thread0_inst0_n64_aliasMSR: 2100_A200h
Init: _ccd2_lthree0_core0_thread1_inst0_n65_aliasMSR: 2100_A300h
Init: _ccd2_lthree0_core1_thread0_inst0_n66_aliasMSR: 2102_A200h
Init: _ccd2_lthree0_core1_thread1_inst0_n67_aliasMSR: 2102_A300h
Init: _ccd2_lthree0_core2_thread0_inst0_n68_aliasMSR: 2104_A200h
Init: _ccd2_lthree0_core2_thread1_inst0_n69_aliasMSR: 2104_A300h
Init: _ccd2_lthree0_core3_thread0_inst0_n70_aliasMSR: 2106_A200h
Init: _ccd2_lthree0_core3_thread1_inst0_n71_aliasMSR: 2106_A300h
Init: _ccd2_lthree0_core4_thread0_inst0_n72_aliasMSR: 2108_A200h
Init: _ccd2_lthree0_core4_thread1_inst0_n73_aliasMSR: 2108_A300h
Init: _ccd2_lthree0_core5_thread0_inst0_n74_aliasMSR: 210A_A200h
Init: _ccd2_lthree0_core5_thread1_inst0_n75_aliasMSR: 210A_A300h
Init: _ccd2_lthree0_core6_thread0_inst0_n76_aliasMSR: 210C_A200h
Init: _ccd2_lthree0_core6_thread1_inst0_n77_aliasMSR: 210C_A300h
Init: _ccd2_lthree0_core7_thread0_inst0_n78_aliasMSR: 210E_A200h
Init: _ccd2_lthree0_core7_thread1_inst0_n79_aliasMSR: 210E_A300h
Init: _ccd2_lthree0_core8_thread0_inst0_n80_aliasMSR: 2110_A200h
Init: _ccd2_lthree0_core8_thread1_inst0_n81_aliasMSR: 2110_A300h
Init: _ccd2_lthree0_core9_thread0_inst0_n82_aliasMSR: 2112_A200h
Init: _ccd2_lthree0_core9_thread1_inst0_n83_aliasMSR: 2112_A300h
Init: _ccd2_lthree0_core10_thread0_inst0_n84_aliasMSR: 2114_A200h
Init: _ccd2_lthree0_core10_thread1_inst0_n85_aliasMSR: 2114_A300h
Init: _ccd2_lthree0_core11_thread0_inst0_n86_aliasMSR: 2116_A200h
Init: _ccd2_lthree0_core11_thread1_inst0_n87_aliasMSR: 2116_A300h
Init: _ccd2_lthree0_core12_thread0_inst0_n88_aliasMSR: 2118_A200h
Init: _ccd2_lthree0_core12_thread1_inst0_n89_aliasMSR: 2118_A300h
Init: _ccd2_lthree0_core13_thread0_inst0_n90_aliasMSR: 211A_A200h
Init: _ccd2_lthree0_core13_thread1_inst0_n91_aliasMSR: 211A_A300h
Init: _ccd2_lthree0_core14_thread0_inst0_n92_aliasMSR: 211C_A200h
Init: _ccd2_lthree0_core14_thread1_inst0_n93_aliasMSR: 211C_A300h
Init: _ccd2_lthree0_core15_thread0_inst0_n94_aliasMSR: 211E_A200h
Init: _ccd2_lthree0_core15_thread1_inst0_n95_aliasMSR: 211E_A300h
Init: _ccd3_lthree0_core0_thread0_inst0_n96_aliasMSR: 2180_A200h
Init: _ccd3_lthree0_core0_thread1_inst0_n97_aliasMSR: 2180_A300h
Init: _ccd3_lthree0_core1_thread0_inst0_n98_aliasMSR: 2182_A200h
Init: _ccd3_lthree0_core1_thread1_inst0_n99_aliasMSR: 2182_A300h
Init: _ccd3_lthree0_core2_thread0_inst0_n100_aliasMSR: 2184_A200h
Init: _ccd3_lthree0_core2_thread1_inst0_n101_aliasMSR: 2184_A300h
Init: _ccd3_lthree0_core3_thread0_inst0_n102_aliasMSR: 2186_A200h
Init: _ccd3_lthree0_core3_thread1_inst0_n103_aliasMSR: 2186_A300h
Init: _ccd3_lthree0_core4_thread0_inst0_n104_aliasMSR: 2188_A200h
Init: _ccd3_lthree0_core4_thread1_inst0_n105_aliasMSR: 2188_A300h
Init: _ccd3_lthree0_core5_thread0_inst0_n106_aliasMSR: 218A_A200h
Init: _ccd3_lthree0_core5_thread1_inst0_n107_aliasMSR: 218A_A300h
Init: _ccd3_lthree0_core6_thread0_inst0_n108_aliasMSR: 218C_A200h
Init: _ccd3_lthree0_core6_thread1_inst0_n109_aliasMSR: 218C_A300h
Init: _ccd3_lthree0_core7_thread0_inst0_n110_aliasMSR: 218E_A200h
Init: _ccd3_lthree0_core7_thread1_inst0_n111_aliasMSR: 218E_A300h
Init: _ccd3_lthree0_core8_thread0_inst0_n112_aliasMSR: 2190_A200h
Init: _ccd3_lthree0_core8_thread1_inst0_n113_aliasMSR: 2190_A300h

Init: _ccd3_lthree0_core9_thread0_inst0_n114_aliasMSR: 2192_A200h
Init: _ccd3_lthree0_core9_thread1_inst0_n115_aliasMSR: 2192_A300h
Init: _ccd3_lthree0_core10_thread0_inst0_n116_aliasMSR: 2194_A200h
Init: _ccd3_lthree0_core10_thread1_inst0_n117_aliasMSR: 2194_A300h
Init: _ccd3_lthree0_core11_thread0_inst0_n118_aliasMSR: 2196_A200h
Init: _ccd3_lthree0_core11_thread1_inst0_n119_aliasMSR: 2196_A300h
Init: _ccd3_lthree0_core12_thread0_inst0_n120_aliasMSR: 2198_A200h
Init: _ccd3_lthree0_core12_thread1_inst0_n121_aliasMSR: 2198_A300h
Init: _ccd3_lthree0_core13_thread0_inst0_n122_aliasMSR: 219A_A200h
Init: _ccd3_lthree0_core13_thread1_inst0_n123_aliasMSR: 219A_A300h
Init: _ccd3_lthree0_core14_thread0_inst0_n124_aliasMSR: 219C_A200h
Init: _ccd3_lthree0_core14_thread1_inst0_n125_aliasMSR: 219C_A300h
Init: _ccd3_lthree0_core15_thread0_inst0_n126_aliasMSR: 219E_A200h
Init: _ccd3_lthree0_core15_thread1_inst0_n127_aliasMSR: 219E_A300h
Init: _ccd4_lthree0_core0_thread0_inst0_n128_aliasMSR: 2200_A200h
Init: _ccd4_lthree0_core0_thread1_inst0_n129_aliasMSR: 2200_A300h
Init: _ccd4_lthree0_core1_thread0_inst0_n130_aliasMSR: 2202_A200h
Init: _ccd4_lthree0_core1_thread1_inst0_n131_aliasMSR: 2202_A300h
Init: _ccd4_lthree0_core2_thread0_inst0_n132_aliasMSR: 2204_A200h
Init: _ccd4_lthree0_core2_thread1_inst0_n133_aliasMSR: 2204_A300h
Init: _ccd4_lthree0_core3_thread0_inst0_n134_aliasMSR: 2206_A200h
Init: _ccd4_lthree0_core3_thread1_inst0_n135_aliasMSR: 2206_A300h
Init: _ccd4_lthree0_core4_thread0_inst0_n136_aliasMSR: 2208_A200h
Init: _ccd4_lthree0_core4_thread1_inst0_n137_aliasMSR: 2208_A300h
Init: _ccd4_lthree0_core5_thread0_inst0_n138_aliasMSR: 220A_A200h
Init: _ccd4_lthree0_core5_thread1_inst0_n139_aliasMSR: 220A_A300h
Init: _ccd4_lthree0_core6_thread0_inst0_n140_aliasMSR: 220C_A200h
Init: _ccd4_lthree0_core6_thread1_inst0_n141_aliasMSR: 220C_A300h
Init: _ccd4_lthree0_core7_thread0_inst0_n142_aliasMSR: 220E_A200h
Init: _ccd4_lthree0_core7_thread1_inst0_n143_aliasMSR: 220E_A300h
Init: _ccd4_lthree0_core8_thread0_inst0_n144_aliasMSR: 2210_A200h
Init: _ccd4_lthree0_core8_thread1_inst0_n145_aliasMSR: 2210_A300h
Init: _ccd4_lthree0_core9_thread0_inst0_n146_aliasMSR: 2212_A200h
Init: _ccd4_lthree0_core9_thread1_inst0_n147_aliasMSR: 2212_A300h
Init: _ccd4_lthree0_core10_thread0_inst0_n148_aliasMSR: 2214_A200h
Init: _ccd4_lthree0_core10_thread1_inst0_n149_aliasMSR: 2214_A300h
Init: _ccd4_lthree0_core11_thread0_inst0_n150_aliasMSR: 2216_A200h
Init: _ccd4_lthree0_core11_thread1_inst0_n151_aliasMSR: 2216_A300h
Init: _ccd4_lthree0_core12_thread0_inst0_n152_aliasMSR: 2218_A200h
Init: _ccd4_lthree0_core12_thread1_inst0_n153_aliasMSR: 2218_A300h
Init: _ccd4_lthree0_core13_thread0_inst0_n154_aliasMSR: 221A_A200h
Init: _ccd4_lthree0_core13_thread1_inst0_n155_aliasMSR: 221A_A300h
Init: _ccd4_lthree0_core14_thread0_inst0_n156_aliasMSR: 221C_A200h
Init: _ccd4_lthree0_core14_thread1_inst0_n157_aliasMSR: 221C_A300h
Init: _ccd4_lthree0_core15_thread0_inst0_n158_aliasMSR: 221E_A200h
Init: _ccd4_lthree0_core15_thread1_inst0_n159_aliasMSR: 221E_A300h
Init: _ccd5_lthree0_core0_thread0_inst0_n160_aliasMSR: 2280_A200h
Init: _ccd5_lthree0_core0_thread1_inst0_n161_aliasMSR: 2280_A300h
Init: _ccd5_lthree0_core1_thread0_inst0_n162_aliasMSR: 2282_A200h
Init: _ccd5_lthree0_core1_thread1_inst0_n163_aliasMSR: 2282_A300h
Init: _ccd5_lthree0_core2_thread0_inst0_n164_aliasMSR: 2284_A200h
Init: _ccd5_lthree0_core2_thread1_inst0_n165_aliasMSR: 2284_A300h
Init: _ccd5_lthree0_core3_thread0_inst0_n166_aliasMSR: 2286_A200h
Init: _ccd5_lthree0_core3_thread1_inst0_n167_aliasMSR: 2286_A300h
Init: _ccd5_lthree0_core4_thread0_inst0_n168_aliasMSR: 2288_A200h
Init: _ccd5_lthree0_core4_thread1_inst0_n169_aliasMSR: 2288_A300h
Init: _ccd5_lthree0_core5_thread0_inst0_n170_aliasMSR: 228A_A200h
Init: _ccd5_lthree0_core5_thread1_inst0_n171_aliasMSR: 228A_A300h
Init: _ccd5_lthree0_core6_thread0_inst0_n172_aliasMSR: 228C_A200h
Init: _ccd5_lthree0_core6_thread1_inst0_n173_aliasMSR: 228C_A300h
Init: _ccd5_lthree0_core7_thread0_inst0_n174_aliasMSR: 228E_A200h
Init: _ccd5_lthree0_core7_thread1_inst0_n175_aliasMSR: 228E_A300h
Init: _ccd5_lthree0_core8_thread0_inst0_n176_aliasMSR: 2290_A200h
Init: _ccd5_lthree0_core8_thread1_inst0_n177_aliasMSR: 2290_A300h
Init: _ccd5_lthree0_core9_thread0_inst0_n178_aliasMSR: 2292_A200h

Init: _ccd5_lthree0_core9_thread1_inst0_n179_aliasMSR: 2292_A300h
Init: _ccd5_lthree0_core10_thread0_inst0_n180_aliasMSR: 2294_A200h
Init: _ccd5_lthree0_core10_thread1_inst0_n181_aliasMSR: 2294_A300h
Init: _ccd5_lthree0_core11_thread0_inst0_n182_aliasMSR: 2296_A200h
Init: _ccd5_lthree0_core11_thread1_inst0_n183_aliasMSR: 2296_A300h
Init: _ccd5_lthree0_core12_thread0_inst0_n184_aliasMSR: 2298_A200h
Init: _ccd5_lthree0_core12_thread1_inst0_n185_aliasMSR: 2298_A300h
Init: _ccd5_lthree0_core13_thread0_inst0_n186_aliasMSR: 229A_A200h
Init: _ccd5_lthree0_core13_thread1_inst0_n187_aliasMSR: 229A_A300h
Init: _ccd5_lthree0_core14_thread0_inst0_n188_aliasMSR: 229C_A200h
Init: _ccd5_lthree0_core14_thread1_inst0_n189_aliasMSR: 229C_A300h
Init: _ccd5_lthree0_core15_thread0_inst0_n190_aliasMSR: 229E_A200h
Init: _ccd5_lthree0_core15_thread1_inst0_n191_aliasMSR: 229E_A300h
Init: _ccd6_lthree0_core0_thread0_inst0_n192_aliasMSR: 2300_A200h
Init: _ccd6_lthree0_core0_thread1_inst0_n193_aliasMSR: 2300_A300h
Init: _ccd6_lthree0_core1_thread0_inst0_n194_aliasMSR: 2302_A200h
Init: _ccd6_lthree0_core1_thread1_inst0_n195_aliasMSR: 2302_A300h
Init: _ccd6_lthree0_core2_thread0_inst0_n196_aliasMSR: 2304_A200h
Init: _ccd6_lthree0_core2_thread1_inst0_n197_aliasMSR: 2304_A300h
Init: _ccd6_lthree0_core3_thread0_inst0_n198_aliasMSR: 2306_A200h
Init: _ccd6_lthree0_core3_thread1_inst0_n199_aliasMSR: 2306_A300h
Init: _ccd6_lthree0_core4_thread0_inst0_n200_aliasMSR: 2308_A200h
Init: _ccd6_lthree0_core4_thread1_inst0_n201_aliasMSR: 2308_A300h
Init: _ccd6_lthree0_core5_thread0_inst0_n202_aliasMSR: 230A_A200h
Init: _ccd6_lthree0_core5_thread1_inst0_n203_aliasMSR: 230A_A300h
Init: _ccd6_lthree0_core6_thread0_inst0_n204_aliasMSR: 230C_A200h
Init: _ccd6_lthree0_core6_thread1_inst0_n205_aliasMSR: 230C_A300h
Init: _ccd6_lthree0_core7_thread0_inst0_n206_aliasMSR: 230E_A200h
Init: _ccd6_lthree0_core7_thread1_inst0_n207_aliasMSR: 230E_A300h
Init: _ccd6_lthree0_core8_thread0_inst0_n208_aliasMSR: 2310_A200h
Init: _ccd6_lthree0_core8_thread1_inst0_n209_aliasMSR: 2310_A300h
Init: _ccd6_lthree0_core9_thread0_inst0_n210_aliasMSR: 2312_A200h
Init: _ccd6_lthree0_core9_thread1_inst0_n211_aliasMSR: 2312_A300h
Init: _ccd6_lthree0_core10_thread0_inst0_n212_aliasMSR: 2314_A200h
Init: _ccd6_lthree0_core10_thread1_inst0_n213_aliasMSR: 2314_A300h
Init: _ccd6_lthree0_core11_thread0_inst0_n214_aliasMSR: 2316_A200h
Init: _ccd6_lthree0_core11_thread1_inst0_n215_aliasMSR: 2316_A300h
Init: _ccd6_lthree0_core12_thread0_inst0_n216_aliasMSR: 2318_A200h
Init: _ccd6_lthree0_core12_thread1_inst0_n217_aliasMSR: 2318_A300h
Init: _ccd6_lthree0_core13_thread0_inst0_n218_aliasMSR: 231A_A200h
Init: _ccd6_lthree0_core13_thread1_inst0_n219_aliasMSR: 231A_A300h
Init: _ccd6_lthree0_core14_thread0_inst0_n220_aliasMSR: 231C_A200h
Init: _ccd6_lthree0_core14_thread1_inst0_n221_aliasMSR: 231C_A300h
Init: _ccd6_lthree0_core15_thread0_inst0_n222_aliasMSR: 231E_A200h
Init: _ccd6_lthree0_core15_thread1_inst0_n223_aliasMSR: 231E_A300h
Init: _ccd7_lthree0_core0_thread0_inst0_n224_aliasMSR: 2380_A200h
Init: _ccd7_lthree0_core0_thread1_inst0_n225_aliasMSR: 2380_A300h
Init: _ccd7_lthree0_core1_thread0_inst0_n226_aliasMSR: 2382_A200h
Init: _ccd7_lthree0_core1_thread1_inst0_n227_aliasMSR: 2382_A300h
Init: _ccd7_lthree0_core2_thread0_inst0_n228_aliasMSR: 2384_A200h
Init: _ccd7_lthree0_core2_thread1_inst0_n229_aliasMSR: 2384_A300h
Init: _ccd7_lthree0_core3_thread0_inst0_n230_aliasMSR: 2386_A200h
Init: _ccd7_lthree0_core3_thread1_inst0_n231_aliasMSR: 2386_A300h
Init: _ccd7_lthree0_core4_thread0_inst0_n232_aliasMSR: 2388_A200h
Init: _ccd7_lthree0_core4_thread1_inst0_n233_aliasMSR: 2388_A300h
Init: _ccd7_lthree0_core5_thread0_inst0_n234_aliasMSR: 238A_A200h
Init: _ccd7_lthree0_core5_thread1_inst0_n235_aliasMSR: 238A_A300h
Init: _ccd7_lthree0_core6_thread0_inst0_n236_aliasMSR: 238C_A200h
Init: _ccd7_lthree0_core6_thread1_inst0_n237_aliasMSR: 238C_A300h
Init: _ccd7_lthree0_core7_thread0_inst0_n238_aliasMSR: 238E_A200h
Init: _ccd7_lthree0_core7_thread1_inst0_n239_aliasMSR: 238E_A300h
Init: _ccd7_lthree0_core8_thread0_inst0_n240_aliasMSR: 2390_A200h
Init: _ccd7_lthree0_core8_thread1_inst0_n241_aliasMSR: 2390_A300h
Init: _ccd7_lthree0_core9_thread0_inst0_n242_aliasMSR: 2392_A200h
Init: _ccd7_lthree0_core9_thread1_inst0_n243_aliasMSR: 2392_A300h

Init: _ccd7_lthree0_core10_thread0_inst0_n244_aliasMSR: 2394_A200h
Init: _ccd7_lthree0_core10_thread1_inst0_n245_aliasMSR: 2394_A300h
Init: _ccd7_lthree0_core11_thread0_inst0_n246_aliasMSR: 2396_A200h
Init: _ccd7_lthree0_core11_thread1_inst0_n247_aliasMSR: 2396_A300h
Init: _ccd7_lthree0_core12_thread0_inst0_n248_aliasMSR: 2398_A200h
Init: _ccd7_lthree0_core12_thread1_inst0_n249_aliasMSR: 2398_A300h
Init: _ccd7_lthree0_core13_thread0_inst0_n250_aliasMSR: 239A_A200h
Init: _ccd7_lthree0_core13_thread1_inst0_n251_aliasMSR: 239A_A300h
Init: _ccd7_lthree0_core14_thread0_inst0_n252_aliasMSR: 239C_A200h
Init: _ccd7_lthree0_core14_thread1_inst0_n253_aliasMSR: 239C_A300h
Init: _ccd7_lthree0_core15_thread0_inst0_n254_aliasMSR: 239E_A200h
Init: _ccd7_lthree0_core15_thread1_inst0_n255_aliasMSR: 239E_A300h
Init: _ccd8_lthree0_core0_thread0_inst0_n256_aliasMSR: 2400_A200h
Init: _ccd8_lthree0_core0_thread1_inst0_n257_aliasMSR: 2400_A300h
Init: _ccd8_lthree0_core1_thread0_inst0_n258_aliasMSR: 2402_A200h
Init: _ccd8_lthree0_core1_thread1_inst0_n259_aliasMSR: 2402_A300h
Init: _ccd8_lthree0_core2_thread0_inst0_n260_aliasMSR: 2404_A200h
Init: _ccd8_lthree0_core2_thread1_inst0_n261_aliasMSR: 2404_A300h
Init: _ccd8_lthree0_core3_thread0_inst0_n262_aliasMSR: 2406_A200h
Init: _ccd8_lthree0_core3_thread1_inst0_n263_aliasMSR: 2406_A300h
Init: _ccd8_lthree0_core4_thread0_inst0_n264_aliasMSR: 2408_A200h
Init: _ccd8_lthree0_core4_thread1_inst0_n265_aliasMSR: 2408_A300h
Init: _ccd8_lthree0_core5_thread0_inst0_n266_aliasMSR: 240A_A200h
Init: _ccd8_lthree0_core5_thread1_inst0_n267_aliasMSR: 240A_A300h
Init: _ccd8_lthree0_core6_thread0_inst0_n268_aliasMSR: 240C_A200h
Init: _ccd8_lthree0_core6_thread1_inst0_n269_aliasMSR: 240C_A300h
Init: _ccd8_lthree0_core7_thread0_inst0_n270_aliasMSR: 240E_A200h
Init: _ccd8_lthree0_core7_thread1_inst0_n271_aliasMSR: 240E_A300h
Init: _ccd8_lthree0_core8_thread0_inst0_n272_aliasMSR: 2410_A200h
Init: _ccd8_lthree0_core8_thread1_inst0_n273_aliasMSR: 2410_A300h
Init: _ccd8_lthree0_core9_thread0_inst0_n274_aliasMSR: 2412_A200h
Init: _ccd8_lthree0_core9_thread1_inst0_n275_aliasMSR: 2412_A300h
Init: _ccd8_lthree0_core10_thread0_inst0_n276_aliasMSR: 2414_A200h
Init: _ccd8_lthree0_core10_thread1_inst0_n277_aliasMSR: 2414_A300h
Init: _ccd8_lthree0_core11_thread0_inst0_n278_aliasMSR: 2416_A200h
Init: _ccd8_lthree0_core11_thread1_inst0_n279_aliasMSR: 2416_A300h
Init: _ccd8_lthree0_core12_thread0_inst0_n280_aliasMSR: 2418_A200h
Init: _ccd8_lthree0_core12_thread1_inst0_n281_aliasMSR: 2418_A300h
Init: _ccd8_lthree0_core13_thread0_inst0_n282_aliasMSR: 241A_A200h
Init: _ccd8_lthree0_core13_thread1_inst0_n283_aliasMSR: 241A_A300h
Init: _ccd8_lthree0_core14_thread0_inst0_n284_aliasMSR: 241C_A200h
Init: _ccd8_lthree0_core14_thread1_inst0_n285_aliasMSR: 241C_A300h
Init: _ccd8_lthree0_core15_thread0_inst0_n286_aliasMSR: 241E_A200h
Init: _ccd8_lthree0_core15_thread1_inst0_n287_aliasMSR: 241E_A300h
Init: _ccd9_lthree0_core0_thread0_inst0_n288_aliasMSR: 2480_A200h
Init: _ccd9_lthree0_core0_thread1_inst0_n289_aliasMSR: 2480_A300h
Init: _ccd9_lthree0_core1_thread0_inst0_n290_aliasMSR: 2482_A200h
Init: _ccd9_lthree0_core1_thread1_inst0_n291_aliasMSR: 2482_A300h
Init: _ccd9_lthree0_core2_thread0_inst0_n292_aliasMSR: 2484_A200h
Init: _ccd9_lthree0_core2_thread1_inst0_n293_aliasMSR: 2484_A300h
Init: _ccd9_lthree0_core3_thread0_inst0_n294_aliasMSR: 2486_A200h
Init: _ccd9_lthree0_core3_thread1_inst0_n295_aliasMSR: 2486_A300h
Init: _ccd9_lthree0_core4_thread0_inst0_n296_aliasMSR: 2488_A200h
Init: _ccd9_lthree0_core4_thread1_inst0_n297_aliasMSR: 2488_A300h
Init: _ccd9_lthree0_core5_thread0_inst0_n298_aliasMSR: 248A_A200h
Init: _ccd9_lthree0_core5_thread1_inst0_n299_aliasMSR: 248A_A300h
Init: _ccd9_lthree0_core6_thread0_inst0_n300_aliasMSR: 248C_A200h
Init: _ccd9_lthree0_core6_thread1_inst0_n301_aliasMSR: 248C_A300h
Init: _ccd9_lthree0_core7_thread0_inst0_n302_aliasMSR: 248E_A200h
Init: _ccd9_lthree0_core7_thread1_inst0_n303_aliasMSR: 248E_A300h
Init: _ccd9_lthree0_core8_thread0_inst0_n304_aliasMSR: 2490_A200h
Init: _ccd9_lthree0_core8_thread1_inst0_n305_aliasMSR: 2490_A300h
Init: _ccd9_lthree0_core9_thread0_inst0_n306_aliasMSR: 2492_A200h
Init: _ccd9_lthree0_core9_thread1_inst0_n307_aliasMSR: 2492_A300h
Init: _ccd9_lthree0_core10_thread0_inst0_n308_aliasMSR: 2494_A200h

Init: _ccd9_lthree0_core10_thread1_inst0_n309_aliasMSR: 2494_A300h
Init: _ccd9_lthree0_core11_thread0_inst0_n310_aliasMSR: 2496_A200h
Init: _ccd9_lthree0_core11_thread1_inst0_n311_aliasMSR: 2496_A300h
Init: _ccd9_lthree0_core12_thread0_inst0_n312_aliasMSR: 2498_A200h
Init: _ccd9_lthree0_core12_thread1_inst0_n313_aliasMSR: 2498_A300h
Init: _ccd9_lthree0_core13_thread0_inst0_n314_aliasMSR: 249A_A200h
Init: _ccd9_lthree0_core13_thread1_inst0_n315_aliasMSR: 249A_A300h
Init: _ccd9_lthree0_core14_thread0_inst0_n316_aliasMSR: 249C_A200h
Init: _ccd9_lthree0_core14_thread1_inst0_n317_aliasMSR: 249C_A300h
Init: _ccd9_lthree0_core15_thread0_inst0_n318_aliasMSR: 249E_A200h
Init: _ccd9_lthree0_core15_thread1_inst0_n319_aliasMSR: 249E_A300h
Init: _ccd10_lthree0_core0_thread0_inst0_n320_aliasMSR: 2500_A200h
Init: _ccd10_lthree0_core0_thread1_inst0_n321_aliasMSR: 2500_A300h
Init: _ccd10_lthree0_core1_thread0_inst0_n322_aliasMSR: 2502_A200h
Init: _ccd10_lthree0_core1_thread1_inst0_n323_aliasMSR: 2502_A300h
Init: _ccd10_lthree0_core2_thread0_inst0_n324_aliasMSR: 2504_A200h
Init: _ccd10_lthree0_core2_thread1_inst0_n325_aliasMSR: 2504_A300h
Init: _ccd10_lthree0_core3_thread0_inst0_n326_aliasMSR: 2506_A200h
Init: _ccd10_lthree0_core3_thread1_inst0_n327_aliasMSR: 2506_A300h
Init: _ccd10_lthree0_core4_thread0_inst0_n328_aliasMSR: 2508_A200h
Init: _ccd10_lthree0_core4_thread1_inst0_n329_aliasMSR: 2508_A300h
Init: _ccd10_lthree0_core5_thread0_inst0_n330_aliasMSR: 250A_A200h
Init: _ccd10_lthree0_core5_thread1_inst0_n331_aliasMSR: 250A_A300h
Init: _ccd10_lthree0_core6_thread0_inst0_n332_aliasMSR: 250C_A200h
Init: _ccd10_lthree0_core6_thread1_inst0_n333_aliasMSR: 250C_A300h
Init: _ccd10_lthree0_core7_thread0_inst0_n334_aliasMSR: 250E_A200h
Init: _ccd10_lthree0_core7_thread1_inst0_n335_aliasMSR: 250E_A300h
Init: _ccd10_lthree0_core8_thread0_inst0_n336_aliasMSR: 2510_A200h
Init: _ccd10_lthree0_core8_thread1_inst0_n337_aliasMSR: 2510_A300h
Init: _ccd10_lthree0_core9_thread0_inst0_n338_aliasMSR: 2512_A200h
Init: _ccd10_lthree0_core9_thread1_inst0_n339_aliasMSR: 2512_A300h
Init: _ccd10_lthree0_core10_thread0_inst0_n340_aliasMSR: 2514_A200h
Init: _ccd10_lthree0_core10_thread1_inst0_n341_aliasMSR: 2514_A300h
Init: _ccd10_lthree0_core11_thread0_inst0_n342_aliasMSR: 2516_A200h
Init: _ccd10_lthree0_core11_thread1_inst0_n343_aliasMSR: 2516_A300h
Init: _ccd10_lthree0_core12_thread0_inst0_n344_aliasMSR: 2518_A200h
Init: _ccd10_lthree0_core12_thread1_inst0_n345_aliasMSR: 2518_A300h
Init: _ccd10_lthree0_core13_thread0_inst0_n346_aliasMSR: 251A_A200h
Init: _ccd10_lthree0_core13_thread1_inst0_n347_aliasMSR: 251A_A300h
Init: _ccd10_lthree0_core14_thread0_inst0_n348_aliasMSR: 251C_A200h
Init: _ccd10_lthree0_core14_thread1_inst0_n349_aliasMSR: 251C_A300h
Init: _ccd10_lthree0_core15_thread0_inst0_n350_aliasMSR: 251E_A200h
Init: _ccd10_lthree0_core15_thread1_inst0_n351_aliasMSR: 251E_A300h
Init: _ccd11_lthree0_core0_thread0_inst0_n352_aliasMSR: 2580_A200h
Init: _ccd11_lthree0_core0_thread1_inst0_n353_aliasMSR: 2580_A300h
Init: _ccd11_lthree0_core1_thread0_inst0_n354_aliasMSR: 2582_A200h
Init: _ccd11_lthree0_core1_thread1_inst0_n355_aliasMSR: 2582_A300h
Init: _ccd11_lthree0_core2_thread0_inst0_n356_aliasMSR: 2584_A200h
Init: _ccd11_lthree0_core2_thread1_inst0_n357_aliasMSR: 2584_A300h
Init: _ccd11_lthree0_core3_thread0_inst0_n358_aliasMSR: 2586_A200h
Init: _ccd11_lthree0_core3_thread1_inst0_n359_aliasMSR: 2586_A300h
Init: _ccd11_lthree0_core4_thread0_inst0_n360_aliasMSR: 2588_A200h
Init: _ccd11_lthree0_core4_thread1_inst0_n361_aliasMSR: 2588_A300h
Init: _ccd11_lthree0_core5_thread0_inst0_n362_aliasMSR: 258A_A200h
Init: _ccd11_lthree0_core5_thread1_inst0_n363_aliasMSR: 258A_A300h
Init: _ccd11_lthree0_core6_thread0_inst0_n364_aliasMSR: 258C_A200h
Init: _ccd11_lthree0_core6_thread1_inst0_n365_aliasMSR: 258C_A300h
Init: _ccd11_lthree0_core7_thread0_inst0_n366_aliasMSR: 258E_A200h
Init: _ccd11_lthree0_core7_thread1_inst0_n367_aliasMSR: 258E_A300h
Init: _ccd11_lthree0_core8_thread0_inst0_n368_aliasMSR: 2590_A200h
Init: _ccd11_lthree0_core8_thread1_inst0_n369_aliasMSR: 2590_A300h
Init: _ccd11_lthree0_core9_thread0_inst0_n370_aliasMSR: 2592_A200h
Init: _ccd11_lthree0_core9_thread1_inst0_n371_aliasMSR: 2592_A300h
Init: _ccd11_lthree0_core10_thread0_inst0_n372_aliasMSR: 2594_A200h
Init: _ccd11_lthree0_core10_thread1_inst0_n373_aliasMSR: 2594_A300h

Init: _ccd11_lthree0_core11_thread0_inst0_n374_aliasMSR: 2596_A200h
Init: _ccd11_lthree0_core11_thread1_inst0_n375_aliasMSR: 2596_A300h
Init: _ccd11_lthree0_core12_thread0_inst0_n376_aliasMSR: 2598_A200h
Init: _ccd11_lthree0_core12_thread1_inst0_n377_aliasMSR: 2598_A300h
Init: _ccd11_lthree0_core13_thread0_inst0_n378_aliasMSR: 259A_A200h
Init: _ccd11_lthree0_core13_thread1_inst0_n379_aliasMSR: 259A_A300h
Init: _ccd11_lthree0_core14_thread0_inst0_n380_aliasMSR: 259C_A200h
Init: _ccd11_lthree0_core14_thread1_inst0_n381_aliasMSR: 259C_A300h
Init: _ccd11_lthree0_core15_thread0_inst0_n382_aliasMSR: 259E_A200h
Init: _ccd11_lthree0_core15_thread1_inst0_n383_aliasMSR: 259E_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](#)

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst0_n[383:0]_aliasMSR; MSRC000_2006

Bits	Description
63:32	Syndrom . 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	ErrorPriority . 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	Length . Read-write. Reset: Cold,00h. Specifies the length in bits of the syndrome contained in MCA::LS::MCA_SYND_LS[Syndrome]. Length values greater than 32 (decimal) are interpreted as equal to 32 (decimal). 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	ErrorInformation . Read-write. Reset: Cold,0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 54 [MCA_SYND_LS].

Table 54: 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]	Reserved Thread ID Reserved

	[0]	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] [7:4] [3:0]	2'b11 Reserved Reserved 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst0_n[383:0]_aliasMSR; MSRC000_2008

Bits	Description
63	Val. Read-write. Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	Overflow. 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 overwrite priorities.)
61:59	RESERV4. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	AddrV. Read-write. Reset: Cold,0. 1=MCA::LS::MCA_DEADDR_LS contains address information associated with the error.
57:54	RESERV3. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	SyndV. 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	RESERV2. Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	Deferred. 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	RESERV1. Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	AddrLsb. 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	RESERV0. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	ErrorCodeExt. 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	ErrorCode. 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].

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst0_n[383:0]_aliasMSR; MSRC000_2009

Bits	Description
63:0	ErrorAddr. 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst0_n[383:0]_aliasMSR; MSRC001_0400

Bits	Description
63:27	Reserved.
26	Hwa . Read-write. Reset: 0. Hardware Asserts (HWAs)
25	SystemReadDataErrorWcb . 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	ScbDataNonCacheable . Read-write. Reset: 0. Error on SCB data for non-cacheable DRAM or IO, discovered at data-pull time
23	ScbData1 . Read-write. Reset: 0. Error on SCB data, commit pipe 1, discovered at SCB commit time
22	ScbData0 . Read-write. Reset: 0. Error on SCB data, commit pipe 0, discovered at SCB commit time
21	ScbStateAddr . Read-write. Reset: 0. Error on SCB cacheline state (way and moesi state) or address field
20	L2DataErr . Read-write. Reset: 0. L2 Fill Data error.
19	DcTagErr7 . Read-write. Reset: 0. DC Tag error type 5.
18	DcTagErr3 . Read-write. Reset: 0. DC Tag error type 3.
17	PDC . Read-write. Reset: 0. PDC parity error.
16	L2DTLB . Read-write. Reset: 0. Level 2 TLB parity error.
15	DcTagErr4 . Read-write. Reset: 0. DC Tag error type 4.
14	DcDataErr3 . Read-write. Reset: 0. DC Data error type 3.
13	DcDataErr2 . Read-write. Reset: 0. DC Data error type 2.
12	DcDataErr1 . Read-write. Reset: 0. DC Data error type 1 and poison consumption. MCA_STATUS[Poison] is set on poison consumption from L2/L3.
11	DcTagErr2 . Read-write. Reset: 0. DC Tag error type 2.
10	SystemReadDataErrorMab . 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	SystemReadDataErrorUcode . 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	IntErrTyp2 . Read-write. Reset: 0. Internal error type 2.
7	IntErrTyp1 . Read-write. Reset: 0. Internal error type 1.
6	DcTagErr1 . Read-write. Reset: 0. DC Tag error type 1.
5	DcTagErr6 . Read-write. Reset: 0. DC Tag error type 6.
4	DcTagErr5 . Read-write. Reset: 0. DC Tag error type 5.
3	L1DTLB . Read-write. Reset: 0. Level 1 TLB parity error.
2	MAB . Read-write. Reset: 0. Miss address buffer payload parity error.
1	STQ . Read-write. Reset: 0. Store queue parity error.
0	LDQ . 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](#)

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst0_n[383:0]_aliasMSR; MSRC000_200E

Bits	Description
63:0	Syndrome . 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](#)`_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst0_n[383:0]_aliasMSR; MSRC000_200F`

Bits	Description
63:0	Syndrone. 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.

`_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst1_n[383:0]_aliasMSRLEGACY; MSR0000_0404``_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst1_n[383:0]_aliasMSR; MSRC000_2010`

Bits	Description
63:20	Reserved.
19	RSVD19. Read-write. Reset: 0.
18	CtMceError. Read-write. Reset: 0. CT MCE
17	RSVD17. Read-write. Reset: 0. Reserved. Will never trigger.
16	L2TlbMultiHit. Read-write. Reset: 0. L2-TLB Multi-Hit
15	L1TlbMultiHit. Read-write. Reset: 0. L1-TLB Multi-Hit.
14	HwAssert. Read-write. Reset: 0. Hardware Assertion Error.
13	SystemReadDataError. 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	L2RespPoison. Read-write. Reset: 0. L2 Cache Response Poison Error. Error is the result of consuming poison data.
11	L2BtbMultiHit. Read-write. Reset: 0. L2 BTB Multi-Match Error.
10	L1BtbMultiHit. Read-write. Reset: 0. L1 BTB Multi-Match Error.
9	BpqSnpParT1. Read-write. Reset: 0. BPQ Thread 1 Snoop Parity Error.
8	BpqSnpParT0. Read-write. Reset: 0. BPQ Thread 0 Snoop Parity Error.
7	L2ItlbParity. Read-write. Reset: 0. L2 ITLB Parity Error.
6	L1ItlbParity. Read-write. Reset: 0. L1 ITLB Parity Error.
5	RSVD5. Read-write. Reset: 0. Reserved. Will never trigger.
4	DqParity. Read-write. Reset: 0. Decoupling Queue PhysAddr Parity Error.
3	DataParity. Read-write. Reset: 0. IC Data Array Parity Error.
2	TagParity. Read-write. Reset: 0. IC Full Tag Parity Error.
1	TagMultiHit. Read-write. Reset: 0. IC Microtag or Full Tag Multi-hit Error.
0	OcUtagParity. 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst1_n[383:0]_aliasMSRLEGACY; MSR0000_0405

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst1_n[383:0]_aliasMSR; MSRC000_2011

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. 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	MiscV. 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	AddrV. 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	PCC. 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	ErrCoreIdVal. 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	TCC. 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	RESERV54. 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	SyndV. 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	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. 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	RESERV22. 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	ErrorCodeExt. 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	ErrorCode. 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 55: 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst1_n[383:0]_aliasMSRLEGACY; MSR0000_0406

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst1_n[383:0]_aliasMSR; MSRC000_2012

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::IF::MCA_STATUS_IF. For physical addresses, the most significant bit is given by Core::X86::Cpuid::LongModeInfo[PhysAddrSize].

Table 56: 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst1_n[383:0]_aliasMSRLEGACY; MSR0000_0407

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst1_n[383:0]_aliasMSR; MSRC000_2013

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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	LvtOffset. 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	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn] !MCA::IF::MCA_MISC0_IF[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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::IF::MCA_MISC0_IF[Locked]) ? Read-write : Read-only.
48	Ovrflw. 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	ErrCnt. 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	BlkPtr. 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst1_n[383:0]_aliasMSR; MSRC000_2014

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	McaXEnable. 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	IntPresent. 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	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::IF::MCA_CONFIG_IF[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. 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	DeferredErrorLoggingSupported. 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	McaX. 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst1_n[383:0]_aliasMSR; MSRC000_2015

Bits	Description
63:48	McaType . Read-only. Reset: 0001h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi . 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	HardwareID . Read-only. Reset: 0B0h. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _ccd0_lthree0_core0_thread0_inst1_n0_aliasMSR: 2000_AA00h
	Init: _ccd0_lthree0_core0_thread1_inst1_n1_aliasMSR: 2000_AB00h
	Init: _ccd0_lthree0_core1_thread0_inst1_n2_aliasMSR: 2002_AA00h
	Init: _ccd0_lthree0_core1_thread1_inst1_n3_aliasMSR: 2002_AB00h
	Init: _ccd0_lthree0_core2_thread0_inst1_n4_aliasMSR: 2004_AA00h
	Init: _ccd0_lthree0_core2_thread1_inst1_n5_aliasMSR: 2004_AB00h
	Init: _ccd0_lthree0_core3_thread0_inst1_n6_aliasMSR: 2006_AA00h
	Init: _ccd0_lthree0_core3_thread1_inst1_n7_aliasMSR: 2006_AB00h
	Init: _ccd0_lthree0_core4_thread0_inst1_n8_aliasMSR: 2008_AA00h
	Init: _ccd0_lthree0_core4_thread1_inst1_n9_aliasMSR: 2008_AB00h
	Init: _ccd0_lthree0_core5_thread0_inst1_n10_aliasMSR: 200A_AA00h
	Init: _ccd0_lthree0_core5_thread1_inst1_n11_aliasMSR: 200A_AB00h
	Init: _ccd0_lthree0_core6_thread0_inst1_n12_aliasMSR: 200C_AA00h
	Init: _ccd0_lthree0_core6_thread1_inst1_n13_aliasMSR: 200C_AB00h
	Init: _ccd0_lthree0_core7_thread0_inst1_n14_aliasMSR: 200E_AA00h
	Init: _ccd0_lthree0_core7_thread1_inst1_n15_aliasMSR: 200E_AB00h
	Init: _ccd0_lthree0_core8_thread0_inst1_n16_aliasMSR: 2010_AA00h
	Init: _ccd0_lthree0_core8_thread1_inst1_n17_aliasMSR: 2010_AB00h
	Init: _ccd0_lthree0_core9_thread0_inst1_n18_aliasMSR: 2012_AA00h
	Init: _ccd0_lthree0_core9_thread1_inst1_n19_aliasMSR: 2012_AB00h
	Init: _ccd0_lthree0_core10_thread0_inst1_n20_aliasMSR: 2014_AA00h
	Init: _ccd0_lthree0_core10_thread1_inst1_n21_aliasMSR: 2014_AB00h
	Init: _ccd0_lthree0_core11_thread0_inst1_n22_aliasMSR: 2016_AA00h
	Init: _ccd0_lthree0_core11_thread1_inst1_n23_aliasMSR: 2016_AB00h
	Init: _ccd0_lthree0_core12_thread0_inst1_n24_aliasMSR: 2018_AA00h
	Init: _ccd0_lthree0_core12_thread1_inst1_n25_aliasMSR: 2018_AB00h
	Init: _ccd0_lthree0_core13_thread0_inst1_n26_aliasMSR: 201A_AA00h
	Init: _ccd0_lthree0_core13_thread1_inst1_n27_aliasMSR: 201A_AB00h
	Init: _ccd0_lthree0_core14_thread0_inst1_n28_aliasMSR: 201C_AA00h
	Init: _ccd0_lthree0_core14_thread1_inst1_n29_aliasMSR: 201C_AB00h
	Init: _ccd0_lthree0_core15_thread0_inst1_n30_aliasMSR: 201E_AA00h
	Init: _ccd0_lthree0_core15_thread1_inst1_n31_aliasMSR: 201E_AB00h
	Init: _ccd1_lthree0_core0_thread0_inst1_n32_aliasMSR: 2080_AA00h
	Init: _ccd1_lthree0_core0_thread1_inst1_n33_aliasMSR: 2080_AB00h
	Init: _ccd1_lthree0_core1_thread0_inst1_n34_aliasMSR: 2082_AA00h
	Init: _ccd1_lthree0_core1_thread1_inst1_n35_aliasMSR: 2082_AB00h
	Init: _ccd1_lthree0_core2_thread0_inst1_n36_aliasMSR: 2084_AA00h
	Init: _ccd1_lthree0_core2_thread1_inst1_n37_aliasMSR: 2084_AB00h
	Init: _ccd1_lthree0_core3_thread0_inst1_n38_aliasMSR: 2086_AA00h
	Init: _ccd1_lthree0_core3_thread1_inst1_n39_aliasMSR: 2086_AB00h
	Init: _ccd1_lthree0_core4_thread0_inst1_n40_aliasMSR: 2088_AA00h
	Init: _ccd1_lthree0_core4_thread1_inst1_n41_aliasMSR: 2088_AB00h
	Init: _ccd1_lthree0_core5_thread0_inst1_n42_aliasMSR: 208A_AA00h
	Init: _ccd1_lthree0_core5_thread1_inst1_n43_aliasMSR: 208A_AB00h
	Init: _ccd1_lthree0_core6_thread0_inst1_n44_aliasMSR: 208C_AA00h
	Init: _ccd1_lthree0_core6_thread1_inst1_n45_aliasMSR: 208C_AB00h
	Init: _ccd1_lthree0_core7_thread0_inst1_n46_aliasMSR: 208E_AA00h
	Init: _ccd1_lthree0_core7_thread1_inst1_n47_aliasMSR: 208E_AB00h
	Init: _ccd1_lthree0_core8_thread0_inst1_n48_aliasMSR: 2090_AA00h

Init: _ccd1_lthree0_core8_thread1_inst1_n49_aliasMSR: 2090_AB00h
Init: _ccd1_lthree0_core9_thread0_inst1_n50_aliasMSR: 2092_AA00h
Init: _ccd1_lthree0_core9_thread1_inst1_n51_aliasMSR: 2092_AB00h
Init: _ccd1_lthree0_core10_thread0_inst1_n52_aliasMSR: 2094_AA00h
Init: _ccd1_lthree0_core10_thread1_inst1_n53_aliasMSR: 2094_AB00h
Init: _ccd1_lthree0_core11_thread0_inst1_n54_aliasMSR: 2096_AA00h
Init: _ccd1_lthree0_core11_thread1_inst1_n55_aliasMSR: 2096_AB00h
Init: _ccd1_lthree0_core12_thread0_inst1_n56_aliasMSR: 2098_AA00h
Init: _ccd1_lthree0_core12_thread1_inst1_n57_aliasMSR: 2098_AB00h
Init: _ccd1_lthree0_core13_thread0_inst1_n58_aliasMSR: 209A_AA00h
Init: _ccd1_lthree0_core13_thread1_inst1_n59_aliasMSR: 209A_AB00h
Init: _ccd1_lthree0_core14_thread0_inst1_n60_aliasMSR: 209C_AA00h
Init: _ccd1_lthree0_core14_thread1_inst1_n61_aliasMSR: 209C_AB00h
Init: _ccd1_lthree0_core15_thread0_inst1_n62_aliasMSR: 209E_AA00h
Init: _ccd1_lthree0_core15_thread1_inst1_n63_aliasMSR: 209E_AB00h
Init: _ccd2_lthree0_core0_thread0_inst1_n64_aliasMSR: 2100_AA00h
Init: _ccd2_lthree0_core0_thread1_inst1_n65_aliasMSR: 2100_AB00h
Init: _ccd2_lthree0_core1_thread0_inst1_n66_aliasMSR: 2102_AA00h
Init: _ccd2_lthree0_core1_thread1_inst1_n67_aliasMSR: 2102_AB00h
Init: _ccd2_lthree0_core2_thread0_inst1_n68_aliasMSR: 2104_AA00h
Init: _ccd2_lthree0_core2_thread1_inst1_n69_aliasMSR: 2104_AB00h
Init: _ccd2_lthree0_core3_thread0_inst1_n70_aliasMSR: 2106_AA00h
Init: _ccd2_lthree0_core3_thread1_inst1_n71_aliasMSR: 2106_AB00h
Init: _ccd2_lthree0_core4_thread0_inst1_n72_aliasMSR: 2108_AA00h
Init: _ccd2_lthree0_core4_thread1_inst1_n73_aliasMSR: 2108_AB00h
Init: _ccd2_lthree0_core5_thread0_inst1_n74_aliasMSR: 210A_AA00h
Init: _ccd2_lthree0_core5_thread1_inst1_n75_aliasMSR: 210A_AB00h
Init: _ccd2_lthree0_core6_thread0_inst1_n76_aliasMSR: 210C_AA00h
Init: _ccd2_lthree0_core6_thread1_inst1_n77_aliasMSR: 210C_AB00h
Init: _ccd2_lthree0_core7_thread0_inst1_n78_aliasMSR: 210E_AA00h
Init: _ccd2_lthree0_core7_thread1_inst1_n79_aliasMSR: 210E_AB00h
Init: _ccd2_lthree0_core8_thread0_inst1_n80_aliasMSR: 2110_AA00h
Init: _ccd2_lthree0_core8_thread1_inst1_n81_aliasMSR: 2110_AB00h
Init: _ccd2_lthree0_core9_thread0_inst1_n82_aliasMSR: 2112_AA00h
Init: _ccd2_lthree0_core9_thread1_inst1_n83_aliasMSR: 2112_AB00h
Init: _ccd2_lthree0_core10_thread0_inst1_n84_aliasMSR: 2114_AA00h
Init: _ccd2_lthree0_core10_thread1_inst1_n85_aliasMSR: 2114_AB00h
Init: _ccd2_lthree0_core11_thread0_inst1_n86_aliasMSR: 2116_AA00h
Init: _ccd2_lthree0_core11_thread1_inst1_n87_aliasMSR: 2116_AB00h
Init: _ccd2_lthree0_core12_thread0_inst1_n88_aliasMSR: 2118_AA00h
Init: _ccd2_lthree0_core12_thread1_inst1_n89_aliasMSR: 2118_AB00h
Init: _ccd2_lthree0_core13_thread0_inst1_n90_aliasMSR: 211A_AA00h
Init: _ccd2_lthree0_core13_thread1_inst1_n91_aliasMSR: 211A_AB00h
Init: _ccd2_lthree0_core14_thread0_inst1_n92_aliasMSR: 211C_AA00h
Init: _ccd2_lthree0_core14_thread1_inst1_n93_aliasMSR: 211C_AB00h
Init: _ccd2_lthree0_core15_thread0_inst1_n94_aliasMSR: 211E_AA00h
Init: _ccd2_lthree0_core15_thread1_inst1_n95_aliasMSR: 211E_AB00h
Init: _ccd3_lthree0_core0_thread0_inst1_n96_aliasMSR: 2180_AA00h
Init: _ccd3_lthree0_core0_thread1_inst1_n97_aliasMSR: 2180_AB00h
Init: _ccd3_lthree0_core1_thread0_inst1_n98_aliasMSR: 2182_AA00h
Init: _ccd3_lthree0_core1_thread1_inst1_n99_aliasMSR: 2182_AB00h
Init: _ccd3_lthree0_core2_thread0_inst1_n100_aliasMSR: 2184_AA00h
Init: _ccd3_lthree0_core2_thread1_inst1_n101_aliasMSR: 2184_AB00h
Init: _ccd3_lthree0_core3_thread0_inst1_n102_aliasMSR: 2186_AA00h
Init: _ccd3_lthree0_core3_thread1_inst1_n103_aliasMSR: 2186_AB00h
Init: _ccd3_lthree0_core4_thread0_inst1_n104_aliasMSR: 2188_AA00h
Init: _ccd3_lthree0_core4_thread1_inst1_n105_aliasMSR: 2188_AB00h
Init: _ccd3_lthree0_core5_thread0_inst1_n106_aliasMSR: 218A_AA00h
Init: _ccd3_lthree0_core5_thread1_inst1_n107_aliasMSR: 218A_AB00h
Init: _ccd3_lthree0_core6_thread0_inst1_n108_aliasMSR: 218C_AA00h
Init: _ccd3_lthree0_core6_thread1_inst1_n109_aliasMSR: 218C_AB00h
Init: _ccd3_lthree0_core7_thread0_inst1_n110_aliasMSR: 218E_AA00h
Init: _ccd3_lthree0_core7_thread1_inst1_n111_aliasMSR: 218E_AB00h
Init: _ccd3_lthree0_core8_thread0_inst1_n112_aliasMSR: 2190_AA00h
Init: _ccd3_lthree0_core8_thread1_inst1_n113_aliasMSR: 2190_AB00h

Init: _ccd3_lthree0_core9_thread0_inst1_n114_aliasMSR: 2192_AA00h
Init: _ccd3_lthree0_core9_thread1_inst1_n115_aliasMSR: 2192_AB00h
Init: _ccd3_lthree0_core10_thread0_inst1_n116_aliasMSR: 2194_AA00h
Init: _ccd3_lthree0_core10_thread1_inst1_n117_aliasMSR: 2194_AB00h
Init: _ccd3_lthree0_core11_thread0_inst1_n118_aliasMSR: 2196_AA00h
Init: _ccd3_lthree0_core11_thread1_inst1_n119_aliasMSR: 2196_AB00h
Init: _ccd3_lthree0_core12_thread0_inst1_n120_aliasMSR: 2198_AA00h
Init: _ccd3_lthree0_core12_thread1_inst1_n121_aliasMSR: 2198_AB00h
Init: _ccd3_lthree0_core13_thread0_inst1_n122_aliasMSR: 219A_AA00h
Init: _ccd3_lthree0_core13_thread1_inst1_n123_aliasMSR: 219A_AB00h
Init: _ccd3_lthree0_core14_thread0_inst1_n124_aliasMSR: 219C_AA00h
Init: _ccd3_lthree0_core14_thread1_inst1_n125_aliasMSR: 219C_AB00h
Init: _ccd3_lthree0_core15_thread0_inst1_n126_aliasMSR: 219E_AA00h
Init: _ccd3_lthree0_core15_thread1_inst1_n127_aliasMSR: 219E_AB00h
Init: _ccd4_lthree0_core0_thread0_inst1_n128_aliasMSR: 2200_AA00h
Init: _ccd4_lthree0_core0_thread1_inst1_n129_aliasMSR: 2200_AB00h
Init: _ccd4_lthree0_core1_thread0_inst1_n130_aliasMSR: 2202_AA00h
Init: _ccd4_lthree0_core1_thread1_inst1_n131_aliasMSR: 2202_AB00h
Init: _ccd4_lthree0_core2_thread0_inst1_n132_aliasMSR: 2204_AA00h
Init: _ccd4_lthree0_core2_thread1_inst1_n133_aliasMSR: 2204_AB00h
Init: _ccd4_lthree0_core3_thread0_inst1_n134_aliasMSR: 2206_AA00h
Init: _ccd4_lthree0_core3_thread1_inst1_n135_aliasMSR: 2206_AB00h
Init: _ccd4_lthree0_core4_thread0_inst1_n136_aliasMSR: 2208_AA00h
Init: _ccd4_lthree0_core4_thread1_inst1_n137_aliasMSR: 2208_AB00h
Init: _ccd4_lthree0_core5_thread0_inst1_n138_aliasMSR: 220A_AA00h
Init: _ccd4_lthree0_core5_thread1_inst1_n139_aliasMSR: 220A_AB00h
Init: _ccd4_lthree0_core6_thread0_inst1_n140_aliasMSR: 220C_AA00h
Init: _ccd4_lthree0_core6_thread1_inst1_n141_aliasMSR: 220C_AB00h
Init: _ccd4_lthree0_core7_thread0_inst1_n142_aliasMSR: 220E_AA00h
Init: _ccd4_lthree0_core7_thread1_inst1_n143_aliasMSR: 220E_AB00h
Init: _ccd4_lthree0_core8_thread0_inst1_n144_aliasMSR: 2210_AA00h
Init: _ccd4_lthree0_core8_thread1_inst1_n145_aliasMSR: 2210_AB00h
Init: _ccd4_lthree0_core9_thread0_inst1_n146_aliasMSR: 2212_AA00h
Init: _ccd4_lthree0_core9_thread1_inst1_n147_aliasMSR: 2212_AB00h
Init: _ccd4_lthree0_core10_thread0_inst1_n148_aliasMSR: 2214_AA00h
Init: _ccd4_lthree0_core10_thread1_inst1_n149_aliasMSR: 2214_AB00h
Init: _ccd4_lthree0_core11_thread0_inst1_n150_aliasMSR: 2216_AA00h
Init: _ccd4_lthree0_core11_thread1_inst1_n151_aliasMSR: 2216_AB00h
Init: _ccd4_lthree0_core12_thread0_inst1_n152_aliasMSR: 2218_AA00h
Init: _ccd4_lthree0_core12_thread1_inst1_n153_aliasMSR: 2218_AB00h
Init: _ccd4_lthree0_core13_thread0_inst1_n154_aliasMSR: 221A_AA00h
Init: _ccd4_lthree0_core13_thread1_inst1_n155_aliasMSR: 221A_AB00h
Init: _ccd4_lthree0_core14_thread0_inst1_n156_aliasMSR: 221C_AA00h
Init: _ccd4_lthree0_core14_thread1_inst1_n157_aliasMSR: 221C_AB00h
Init: _ccd4_lthree0_core15_thread0_inst1_n158_aliasMSR: 221E_AA00h
Init: _ccd4_lthree0_core15_thread1_inst1_n159_aliasMSR: 221E_AB00h
Init: _ccd5_lthree0_core0_thread0_inst1_n160_aliasMSR: 2280_AA00h
Init: _ccd5_lthree0_core0_thread1_inst1_n161_aliasMSR: 2280_AB00h
Init: _ccd5_lthree0_core1_thread0_inst1_n162_aliasMSR: 2282_AA00h
Init: _ccd5_lthree0_core1_thread1_inst1_n163_aliasMSR: 2282_AB00h
Init: _ccd5_lthree0_core2_thread0_inst1_n164_aliasMSR: 2284_AA00h
Init: _ccd5_lthree0_core2_thread1_inst1_n165_aliasMSR: 2284_AB00h
Init: _ccd5_lthree0_core3_thread0_inst1_n166_aliasMSR: 2286_AA00h
Init: _ccd5_lthree0_core3_thread1_inst1_n167_aliasMSR: 2286_AB00h
Init: _ccd5_lthree0_core4_thread0_inst1_n168_aliasMSR: 2288_AA00h
Init: _ccd5_lthree0_core4_thread1_inst1_n169_aliasMSR: 2288_AB00h
Init: _ccd5_lthree0_core5_thread0_inst1_n170_aliasMSR: 228A_AA00h
Init: _ccd5_lthree0_core5_thread1_inst1_n171_aliasMSR: 228A_AB00h
Init: _ccd5_lthree0_core6_thread0_inst1_n172_aliasMSR: 228C_AA00h
Init: _ccd5_lthree0_core6_thread1_inst1_n173_aliasMSR: 228C_AB00h
Init: _ccd5_lthree0_core7_thread0_inst1_n174_aliasMSR: 228E_AA00h
Init: _ccd5_lthree0_core7_thread1_inst1_n175_aliasMSR: 228E_AB00h
Init: _ccd5_lthree0_core8_thread0_inst1_n176_aliasMSR: 2290_AA00h
Init: _ccd5_lthree0_core8_thread1_inst1_n177_aliasMSR: 2290_AB00h
Init: _ccd5_lthree0_core9_thread0_inst1_n178_aliasMSR: 2292_AA00h

Init: _ccd5_lthree0_core9_thread1_inst1_n179_aliasMSR: 2292_AB00h
Init: _ccd5_lthree0_core10_thread0_inst1_n180_aliasMSR: 2294_AA00h
Init: _ccd5_lthree0_core10_thread1_inst1_n181_aliasMSR: 2294_AB00h
Init: _ccd5_lthree0_core11_thread0_inst1_n182_aliasMSR: 2296_AA00h
Init: _ccd5_lthree0_core11_thread1_inst1_n183_aliasMSR: 2296_AB00h
Init: _ccd5_lthree0_core12_thread0_inst1_n184_aliasMSR: 2298_AA00h
Init: _ccd5_lthree0_core12_thread1_inst1_n185_aliasMSR: 2298_AB00h
Init: _ccd5_lthree0_core13_thread0_inst1_n186_aliasMSR: 229A_AA00h
Init: _ccd5_lthree0_core13_thread1_inst1_n187_aliasMSR: 229A_AB00h
Init: _ccd5_lthree0_core14_thread0_inst1_n188_aliasMSR: 229C_AA00h
Init: _ccd5_lthree0_core14_thread1_inst1_n189_aliasMSR: 229C_AB00h
Init: _ccd5_lthree0_core15_thread0_inst1_n190_aliasMSR: 229E_AA00h
Init: _ccd5_lthree0_core15_thread1_inst1_n191_aliasMSR: 229E_AB00h
Init: _ccd6_lthree0_core0_thread0_inst1_n192_aliasMSR: 2300_AA00h
Init: _ccd6_lthree0_core0_thread1_inst1_n193_aliasMSR: 2300_AB00h
Init: _ccd6_lthree0_core1_thread0_inst1_n194_aliasMSR: 2302_AA00h
Init: _ccd6_lthree0_core1_thread1_inst1_n195_aliasMSR: 2302_AB00h
Init: _ccd6_lthree0_core2_thread0_inst1_n196_aliasMSR: 2304_AA00h
Init: _ccd6_lthree0_core2_thread1_inst1_n197_aliasMSR: 2304_AB00h
Init: _ccd6_lthree0_core3_thread0_inst1_n198_aliasMSR: 2306_AA00h
Init: _ccd6_lthree0_core3_thread1_inst1_n199_aliasMSR: 2306_AB00h
Init: _ccd6_lthree0_core4_thread0_inst1_n200_aliasMSR: 2308_AA00h
Init: _ccd6_lthree0_core4_thread1_inst1_n201_aliasMSR: 2308_AB00h
Init: _ccd6_lthree0_core5_thread0_inst1_n202_aliasMSR: 230A_AA00h
Init: _ccd6_lthree0_core5_thread1_inst1_n203_aliasMSR: 230A_AB00h
Init: _ccd6_lthree0_core6_thread0_inst1_n204_aliasMSR: 230C_AA00h
Init: _ccd6_lthree0_core6_thread1_inst1_n205_aliasMSR: 230C_AB00h
Init: _ccd6_lthree0_core7_thread0_inst1_n206_aliasMSR: 230E_AA00h
Init: _ccd6_lthree0_core7_thread1_inst1_n207_aliasMSR: 230E_AB00h
Init: _ccd6_lthree0_core8_thread0_inst1_n208_aliasMSR: 2310_AA00h
Init: _ccd6_lthree0_core8_thread1_inst1_n209_aliasMSR: 2310_AB00h
Init: _ccd6_lthree0_core9_thread0_inst1_n210_aliasMSR: 2312_AA00h
Init: _ccd6_lthree0_core9_thread1_inst1_n211_aliasMSR: 2312_AB00h
Init: _ccd6_lthree0_core10_thread0_inst1_n212_aliasMSR: 2314_AA00h
Init: _ccd6_lthree0_core10_thread1_inst1_n213_aliasMSR: 2314_AB00h
Init: _ccd6_lthree0_core11_thread0_inst1_n214_aliasMSR: 2316_AA00h
Init: _ccd6_lthree0_core11_thread1_inst1_n215_aliasMSR: 2316_AB00h
Init: _ccd6_lthree0_core12_thread0_inst1_n216_aliasMSR: 2318_AA00h
Init: _ccd6_lthree0_core12_thread1_inst1_n217_aliasMSR: 2318_AB00h
Init: _ccd6_lthree0_core13_thread0_inst1_n218_aliasMSR: 231A_AA00h
Init: _ccd6_lthree0_core13_thread1_inst1_n219_aliasMSR: 231A_AB00h
Init: _ccd6_lthree0_core14_thread0_inst1_n220_aliasMSR: 231C_AA00h
Init: _ccd6_lthree0_core14_thread1_inst1_n221_aliasMSR: 231C_AB00h
Init: _ccd6_lthree0_core15_thread0_inst1_n222_aliasMSR: 231E_AA00h
Init: _ccd6_lthree0_core15_thread1_inst1_n223_aliasMSR: 231E_AB00h
Init: _ccd7_lthree0_core0_thread0_inst1_n224_aliasMSR: 2380_AA00h
Init: _ccd7_lthree0_core0_thread1_inst1_n225_aliasMSR: 2380_AB00h
Init: _ccd7_lthree0_core1_thread0_inst1_n226_aliasMSR: 2382_AA00h
Init: _ccd7_lthree0_core1_thread1_inst1_n227_aliasMSR: 2382_AB00h
Init: _ccd7_lthree0_core2_thread0_inst1_n228_aliasMSR: 2384_AA00h
Init: _ccd7_lthree0_core2_thread1_inst1_n229_aliasMSR: 2384_AB00h
Init: _ccd7_lthree0_core3_thread0_inst1_n230_aliasMSR: 2386_AA00h
Init: _ccd7_lthree0_core3_thread1_inst1_n231_aliasMSR: 2386_AB00h
Init: _ccd7_lthree0_core4_thread0_inst1_n232_aliasMSR: 2388_AA00h
Init: _ccd7_lthree0_core4_thread1_inst1_n233_aliasMSR: 2388_AB00h
Init: _ccd7_lthree0_core5_thread0_inst1_n234_aliasMSR: 238A_AA00h
Init: _ccd7_lthree0_core5_thread1_inst1_n235_aliasMSR: 238A_AB00h
Init: _ccd7_lthree0_core6_thread0_inst1_n236_aliasMSR: 238C_AA00h
Init: _ccd7_lthree0_core6_thread1_inst1_n237_aliasMSR: 238C_AB00h
Init: _ccd7_lthree0_core7_thread0_inst1_n238_aliasMSR: 238E_AA00h
Init: _ccd7_lthree0_core7_thread1_inst1_n239_aliasMSR: 238E_AB00h
Init: _ccd7_lthree0_core8_thread0_inst1_n240_aliasMSR: 2390_AA00h
Init: _ccd7_lthree0_core8_thread1_inst1_n241_aliasMSR: 2390_AB00h
Init: _ccd7_lthree0_core9_thread0_inst1_n242_aliasMSR: 2392_AA00h
Init: _ccd7_lthree0_core9_thread1_inst1_n243_aliasMSR: 2392_AB00h

Init: _ccd7_lthree0_core10_thread0_inst1_n244_aliasMSR: 2394_AA00h
Init: _ccd7_lthree0_core10_thread1_inst1_n245_aliasMSR: 2394_AB00h
Init: _ccd7_lthree0_core11_thread0_inst1_n246_aliasMSR: 2396_AA00h
Init: _ccd7_lthree0_core11_thread1_inst1_n247_aliasMSR: 2396_AB00h
Init: _ccd7_lthree0_core12_thread0_inst1_n248_aliasMSR: 2398_AA00h
Init: _ccd7_lthree0_core12_thread1_inst1_n249_aliasMSR: 2398_AB00h
Init: _ccd7_lthree0_core13_thread0_inst1_n250_aliasMSR: 239A_AA00h
Init: _ccd7_lthree0_core13_thread1_inst1_n251_aliasMSR: 239A_AB00h
Init: _ccd7_lthree0_core14_thread0_inst1_n252_aliasMSR: 239C_AA00h
Init: _ccd7_lthree0_core14_thread1_inst1_n253_aliasMSR: 239C_AB00h
Init: _ccd7_lthree0_core15_thread0_inst1_n254_aliasMSR: 239E_AA00h
Init: _ccd7_lthree0_core15_thread1_inst1_n255_aliasMSR: 239E_AB00h
Init: _ccd8_lthree0_core0_thread0_inst1_n256_aliasMSR: 2400_AA00h
Init: _ccd8_lthree0_core0_thread1_inst1_n257_aliasMSR: 2400_AB00h
Init: _ccd8_lthree0_core1_thread0_inst1_n258_aliasMSR: 2402_AA00h
Init: _ccd8_lthree0_core1_thread1_inst1_n259_aliasMSR: 2402_AB00h
Init: _ccd8_lthree0_core2_thread0_inst1_n260_aliasMSR: 2404_AA00h
Init: _ccd8_lthree0_core2_thread1_inst1_n261_aliasMSR: 2404_AB00h
Init: _ccd8_lthree0_core3_thread0_inst1_n262_aliasMSR: 2406_AA00h
Init: _ccd8_lthree0_core3_thread1_inst1_n263_aliasMSR: 2406_AB00h
Init: _ccd8_lthree0_core4_thread0_inst1_n264_aliasMSR: 2408_AA00h
Init: _ccd8_lthree0_core4_thread1_inst1_n265_aliasMSR: 2408_AB00h
Init: _ccd8_lthree0_core5_thread0_inst1_n266_aliasMSR: 240A_AA00h
Init: _ccd8_lthree0_core5_thread1_inst1_n267_aliasMSR: 240A_AB00h
Init: _ccd8_lthree0_core6_thread0_inst1_n268_aliasMSR: 240C_AA00h
Init: _ccd8_lthree0_core6_thread1_inst1_n269_aliasMSR: 240C_AB00h
Init: _ccd8_lthree0_core7_thread0_inst1_n270_aliasMSR: 240E_AA00h
Init: _ccd8_lthree0_core7_thread1_inst1_n271_aliasMSR: 240E_AB00h
Init: _ccd8_lthree0_core8_thread0_inst1_n272_aliasMSR: 2410_AA00h
Init: _ccd8_lthree0_core8_thread1_inst1_n273_aliasMSR: 2410_AB00h
Init: _ccd8_lthree0_core9_thread0_inst1_n274_aliasMSR: 2412_AA00h
Init: _ccd8_lthree0_core9_thread1_inst1_n275_aliasMSR: 2412_AB00h
Init: _ccd8_lthree0_core10_thread0_inst1_n276_aliasMSR: 2414_AA00h
Init: _ccd8_lthree0_core10_thread1_inst1_n277_aliasMSR: 2414_AB00h
Init: _ccd8_lthree0_core11_thread0_inst1_n278_aliasMSR: 2416_AA00h
Init: _ccd8_lthree0_core11_thread1_inst1_n279_aliasMSR: 2416_AB00h
Init: _ccd8_lthree0_core12_thread0_inst1_n280_aliasMSR: 2418_AA00h
Init: _ccd8_lthree0_core12_thread1_inst1_n281_aliasMSR: 2418_AB00h
Init: _ccd8_lthree0_core13_thread0_inst1_n282_aliasMSR: 241A_AA00h
Init: _ccd8_lthree0_core13_thread1_inst1_n283_aliasMSR: 241A_AB00h
Init: _ccd8_lthree0_core14_thread0_inst1_n284_aliasMSR: 241C_AA00h
Init: _ccd8_lthree0_core14_thread1_inst1_n285_aliasMSR: 241C_AB00h
Init: _ccd8_lthree0_core15_thread0_inst1_n286_aliasMSR: 241E_AA00h
Init: _ccd8_lthree0_core15_thread1_inst1_n287_aliasMSR: 241E_AB00h
Init: _ccd9_lthree0_core0_thread0_inst1_n288_aliasMSR: 2480_AA00h
Init: _ccd9_lthree0_core0_thread1_inst1_n289_aliasMSR: 2480_AB00h
Init: _ccd9_lthree0_core1_thread0_inst1_n290_aliasMSR: 2482_AA00h
Init: _ccd9_lthree0_core1_thread1_inst1_n291_aliasMSR: 2482_AB00h
Init: _ccd9_lthree0_core2_thread0_inst1_n292_aliasMSR: 2484_AA00h
Init: _ccd9_lthree0_core2_thread1_inst1_n293_aliasMSR: 2484_AB00h
Init: _ccd9_lthree0_core3_thread0_inst1_n294_aliasMSR: 2486_AA00h
Init: _ccd9_lthree0_core3_thread1_inst1_n295_aliasMSR: 2486_AB00h
Init: _ccd9_lthree0_core4_thread0_inst1_n296_aliasMSR: 2488_AA00h
Init: _ccd9_lthree0_core4_thread1_inst1_n297_aliasMSR: 2488_AB00h
Init: _ccd9_lthree0_core5_thread0_inst1_n298_aliasMSR: 248A_AA00h
Init: _ccd9_lthree0_core5_thread1_inst1_n299_aliasMSR: 248A_AB00h
Init: _ccd9_lthree0_core6_thread0_inst1_n300_aliasMSR: 248C_AA00h
Init: _ccd9_lthree0_core6_thread1_inst1_n301_aliasMSR: 248C_AB00h
Init: _ccd9_lthree0_core7_thread0_inst1_n302_aliasMSR: 248E_AA00h
Init: _ccd9_lthree0_core7_thread1_inst1_n303_aliasMSR: 248E_AB00h
Init: _ccd9_lthree0_core8_thread0_inst1_n304_aliasMSR: 2490_AA00h
Init: _ccd9_lthree0_core8_thread1_inst1_n305_aliasMSR: 2490_AB00h
Init: _ccd9_lthree0_core9_thread0_inst1_n306_aliasMSR: 2492_AA00h
Init: _ccd9_lthree0_core9_thread1_inst1_n307_aliasMSR: 2492_AB00h
Init: _ccd9_lthree0_core10_thread0_inst1_n308_aliasMSR: 2494_AA00h

Init: _ccd9_lthree0_core10_thread1_inst1_n309_aliasMSR: 2494_AB00h
Init: _ccd9_lthree0_core11_thread0_inst1_n310_aliasMSR: 2496_AA00h
Init: _ccd9_lthree0_core11_thread1_inst1_n311_aliasMSR: 2496_AB00h
Init: _ccd9_lthree0_core12_thread0_inst1_n312_aliasMSR: 2498_AA00h
Init: _ccd9_lthree0_core12_thread1_inst1_n313_aliasMSR: 2498_AB00h
Init: _ccd9_lthree0_core13_thread0_inst1_n314_aliasMSR: 249A_AA00h
Init: _ccd9_lthree0_core13_thread1_inst1_n315_aliasMSR: 249A_AB00h
Init: _ccd9_lthree0_core14_thread0_inst1_n316_aliasMSR: 249C_AA00h
Init: _ccd9_lthree0_core14_thread1_inst1_n317_aliasMSR: 249C_AB00h
Init: _ccd9_lthree0_core15_thread0_inst1_n318_aliasMSR: 249E_AA00h
Init: _ccd9_lthree0_core15_thread1_inst1_n319_aliasMSR: 249E_AB00h
Init: _ccd10_lthree0_core0_thread0_inst1_n320_aliasMSR: 2500_AA00h
Init: _ccd10_lthree0_core0_thread1_inst1_n321_aliasMSR: 2500_AB00h
Init: _ccd10_lthree0_core1_thread0_inst1_n322_aliasMSR: 2502_AA00h
Init: _ccd10_lthree0_core1_thread1_inst1_n323_aliasMSR: 2502_AB00h
Init: _ccd10_lthree0_core2_thread0_inst1_n324_aliasMSR: 2504_AA00h
Init: _ccd10_lthree0_core2_thread1_inst1_n325_aliasMSR: 2504_AB00h
Init: _ccd10_lthree0_core3_thread0_inst1_n326_aliasMSR: 2506_AA00h
Init: _ccd10_lthree0_core3_thread1_inst1_n327_aliasMSR: 2506_AB00h
Init: _ccd10_lthree0_core4_thread0_inst1_n328_aliasMSR: 2508_AA00h
Init: _ccd10_lthree0_core4_thread1_inst1_n329_aliasMSR: 2508_AB00h
Init: _ccd10_lthree0_core5_thread0_inst1_n330_aliasMSR: 250A_AA00h
Init: _ccd10_lthree0_core5_thread1_inst1_n331_aliasMSR: 250A_AB00h
Init: _ccd10_lthree0_core6_thread0_inst1_n332_aliasMSR: 250C_AA00h
Init: _ccd10_lthree0_core6_thread1_inst1_n333_aliasMSR: 250C_AB00h
Init: _ccd10_lthree0_core7_thread0_inst1_n334_aliasMSR: 250E_AA00h
Init: _ccd10_lthree0_core7_thread1_inst1_n335_aliasMSR: 250E_AB00h
Init: _ccd10_lthree0_core8_thread0_inst1_n336_aliasMSR: 2510_AA00h
Init: _ccd10_lthree0_core8_thread1_inst1_n337_aliasMSR: 2510_AB00h
Init: _ccd10_lthree0_core9_thread0_inst1_n338_aliasMSR: 2512_AA00h
Init: _ccd10_lthree0_core9_thread1_inst1_n339_aliasMSR: 2512_AB00h
Init: _ccd10_lthree0_core10_thread0_inst1_n340_aliasMSR: 2514_AA00h
Init: _ccd10_lthree0_core10_thread1_inst1_n341_aliasMSR: 2514_AB00h
Init: _ccd10_lthree0_core11_thread0_inst1_n342_aliasMSR: 2516_AA00h
Init: _ccd10_lthree0_core11_thread1_inst1_n343_aliasMSR: 2516_AB00h
Init: _ccd10_lthree0_core12_thread0_inst1_n344_aliasMSR: 2518_AA00h
Init: _ccd10_lthree0_core12_thread1_inst1_n345_aliasMSR: 2518_AB00h
Init: _ccd10_lthree0_core13_thread0_inst1_n346_aliasMSR: 251A_AA00h
Init: _ccd10_lthree0_core13_thread1_inst1_n347_aliasMSR: 251A_AB00h
Init: _ccd10_lthree0_core14_thread0_inst1_n348_aliasMSR: 251C_AA00h
Init: _ccd10_lthree0_core14_thread1_inst1_n349_aliasMSR: 251C_AB00h
Init: _ccd10_lthree0_core15_thread0_inst1_n350_aliasMSR: 251E_AA00h
Init: _ccd10_lthree0_core15_thread1_inst1_n351_aliasMSR: 251E_AB00h
Init: _ccd11_lthree0_core0_thread0_inst1_n352_aliasMSR: 2580_AA00h
Init: _ccd11_lthree0_core0_thread1_inst1_n353_aliasMSR: 2580_AB00h
Init: _ccd11_lthree0_core1_thread0_inst1_n354_aliasMSR: 2582_AA00h
Init: _ccd11_lthree0_core1_thread1_inst1_n355_aliasMSR: 2582_AB00h
Init: _ccd11_lthree0_core2_thread0_inst1_n356_aliasMSR: 2584_AA00h
Init: _ccd11_lthree0_core2_thread1_inst1_n357_aliasMSR: 2584_AB00h
Init: _ccd11_lthree0_core3_thread0_inst1_n358_aliasMSR: 2586_AA00h
Init: _ccd11_lthree0_core3_thread1_inst1_n359_aliasMSR: 2586_AB00h
Init: _ccd11_lthree0_core4_thread0_inst1_n360_aliasMSR: 2588_AA00h
Init: _ccd11_lthree0_core4_thread1_inst1_n361_aliasMSR: 2588_AB00h
Init: _ccd11_lthree0_core5_thread0_inst1_n362_aliasMSR: 258A_AA00h
Init: _ccd11_lthree0_core5_thread1_inst1_n363_aliasMSR: 258A_AB00h
Init: _ccd11_lthree0_core6_thread0_inst1_n364_aliasMSR: 258C_AA00h
Init: _ccd11_lthree0_core6_thread1_inst1_n365_aliasMSR: 258C_AB00h
Init: _ccd11_lthree0_core7_thread0_inst1_n366_aliasMSR: 258E_AA00h
Init: _ccd11_lthree0_core7_thread1_inst1_n367_aliasMSR: 258E_AB00h
Init: _ccd11_lthree0_core8_thread0_inst1_n368_aliasMSR: 2590_AA00h
Init: _ccd11_lthree0_core8_thread1_inst1_n369_aliasMSR: 2590_AB00h
Init: _ccd11_lthree0_core9_thread0_inst1_n370_aliasMSR: 2592_AA00h
Init: _ccd11_lthree0_core9_thread1_inst1_n371_aliasMSR: 2592_AB00h
Init: _ccd11_lthree0_core10_thread0_inst1_n372_aliasMSR: 2594_AA00h
Init: _ccd11_lthree0_core10_thread1_inst1_n373_aliasMSR: 2594_AB00h

Init: _ccd11_lthree0_core11_thread0_inst1_n374_aliasMSR: 2596_AA00h
Init: _ccd11_lthree0_core11_thread1_inst1_n375_aliasMSR: 2596_AB00h
Init: _ccd11_lthree0_core12_thread0_inst1_n376_aliasMSR: 2598_AA00h
Init: _ccd11_lthree0_core12_thread1_inst1_n377_aliasMSR: 2598_AB00h
Init: _ccd11_lthree0_core13_thread0_inst1_n378_aliasMSR: 259A_AA00h
Init: _ccd11_lthree0_core13_thread1_inst1_n379_aliasMSR: 259A_AB00h
Init: _ccd11_lthree0_core14_thread0_inst1_n380_aliasMSR: 259C_AA00h
Init: _ccd11_lthree0_core14_thread1_inst1_n381_aliasMSR: 259C_AB00h
Init: _ccd11_lthree0_core15_thread0_inst1_n382_aliasMSR: 259E_AA00h
Init: _ccd11_lthree0_core15_thread1_inst1_n383_aliasMSR: 259E_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](#)

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst1_n[383:0]_aliasMSR; MSRC000_2016

Bits	Description
63:32	Syndrom . 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	ErrorPriority . 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	Length . Read-write, Volatile . Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::IF::MCA_SYND_IF[Syndrome]. Length values greater than 32 (decimal) are interpreted as equal to 32 (decimal). 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	ErrorInformation . 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_IF].

Table 57: 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]	Reserved

	[5:0]	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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst1_n[383:0]_aliasMSR; MSRC001_0401

Bits	Description
63:20	Reserved.
19	RSVD19 . Read-write. Reset: 0.
18	CtMceError . Read-write. Reset: 0. CT MCE
17	RSVD17 . Read-write. Reset: 0. Reserved. Will never trigger.
16	L2TlbMultiHit . Read-write. Reset: 0. L2-TLB Multi-Hit
15	L1TlbMultiHit . Read-write. Reset: 0. L1-TLB Multi-Hit.
14	HwAssert . Read-write. Reset: 0. Hardware Assertion Error.
13	SystemReadDataError . 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	L2RespPoison . Read-write. Reset: 0. L2 Cache Response Poison Error. Error is the result of consuming poison data.
11	L2BtbMultiHit . Read-write. Reset: 0. L2 BTB Multi-Match Error.
10	L1BtbMultiHit . Read-write. Reset: 0. L1 BTB Multi-Match Error.
9	BpqSnpParT1 . Read-write. Reset: 0. BPQ Thread 1 Snoop Parity Error.
8	BpqSnpParT0 . Read-write. Reset: 0. BPQ Thread 0 Snoop Parity Error.
7	L2ItlbParity . Read-write. Reset: 0. L2 ITLB Parity Error.
6	L1ItlbParity . Read-write. Reset: 0. L1 ITLB Parity Error.
5	RSVD5 . Read-write. Reset: 0. Reserved. Will never trigger.
4	DqParity . Read-write. Reset: 0. Decoupling Queue PhysAddr Parity Error.
3	DataParity . Read-write. Reset: 0. IC Data Array Parity Error.
2	TagParity . Read-write. Reset: 0. IC Full Tag Parity Error.
1	TagMultiHit . Read-write. Reset: 0. IC Microtag or Full Tag Multi-hit Error.
0	OcUtagParity . 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](#)[_ccd\[11:0\]_lthree0_core\[15:0\]_thread\[1:0\]_inst1_n\[383:0\]_aliasMSR; MSRC000_201E](#)

Bits	Description
63:0	Syndrom . 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](#)[_ccd\[11:0\]_lthree0_core\[15:0\]_thread\[1:0\]_inst1_n\[383:0\]_aliasMSR; MSRC000_201F](#)

Bits	Description
63:0	Syndrom . 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.

[_ccd\[11:0\]_lthree0_core\[15:0\]_thread\[1:0\]_inst2_n\[383:0\]_aliasMSRLEGACY; MSR0000_0408](#)[_ccd\[11:0\]_lthree0_core\[15:0\]_thread\[1:0\]_inst2_n\[383:0\]_aliasMSR; MSRC000_2020](#)

Bits	Description
63:7	Reserved.
6	Wdt . Read-write. Reset: 0. Reserved
5	StateMachine . Read-write. Reset: 0. Error initiated by programmable state machine.
4	Sdp . Read-write. Reset: 0. SDP Read Response Parity Error
3	Hwa . Read-write. Reset: 0. Hardware Assert Error.
2	Data . Read-write. Reset: 0. L2M Data Array ECC Error.
1	Tag . Read-write. Reset: 0. L2M Tag or State Array ECC Error.
0	MultiHit . 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst2_n[383:0]_aliasMSRLEGACY; MSR0000_0409

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst2_n[383:0]_aliasMSR; MSRC000_2021

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. 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	MiscV. 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	AddrV. 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	PCC. 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	ErrCoreIdVal. 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	TCC. 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	RESERV54. 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	SyndV. 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	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. 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	RESERV22. 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	ErrorCodeExt. 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	ErrorCode. 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_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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst2_n[383:0]_aliasMSRLEGACY; MSR0000_040A

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst2_n[383:0]_aliasMSR; MSRC000_2022

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::L2::MCA_STATUS_L2. For physical addresses, the most significant bit is given by Core::X86::Cpuid::LongModeInfo[PhysAddrSize].

Table 59: 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst2_n[383:0]_aliasMSRLEGACY; MSR0000_040B

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst2_n[383:0]_aliasMSR; MSRC000_2023

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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::L2::MCA_MISC0_L2[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	LvtOffset. 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::L2::MCA_MISC0_L2[Locked]) ? Read-write : Read-only.
51	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msrr::HWCR[McStatusWrEn] !MCA::L2::MCA_MISC0_L2[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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::L2::MCA_MISC0_L2[Locked]) ? Read-write : Read-only.
48	Ovrflw. 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::L2::MCA_MISC0_L2[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	ErrCnt. 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::L2::MCA_MISC0_L2[Locked]) ? Read-write : Read-only.
31:24	BlkPtr. 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst2_n[383:0]_aliasMSR; MSRC000_2024

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	LogDeferredInMcaStat. 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	McaXEnable. 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	IntPresent. 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	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::L2::MCA_CONFIG_L2[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. 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	DeferredErrorLoggingSupported. 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	McaX. 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst2_n[383:0]_aliasMSR; MSRC000_2025

Bits	Description
63:48	McaType . Read-only. Reset: 0002h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi . 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	HardwareID . Read-only. Reset: 0B0h. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _ccd0_lthree0_core0_thread0_inst2_n0_aliasMSR: 2000_8A00h
	Init: _ccd0_lthree0_core0_thread1_inst2_n1_aliasMSR: 2000_8B00h
	Init: _ccd0_lthree0_core1_thread0_inst2_n2_aliasMSR: 2002_8A00h
	Init: _ccd0_lthree0_core1_thread1_inst2_n3_aliasMSR: 2002_8B00h
	Init: _ccd0_lthree0_core2_thread0_inst2_n4_aliasMSR: 2004_8A00h
	Init: _ccd0_lthree0_core2_thread1_inst2_n5_aliasMSR: 2004_8B00h
	Init: _ccd0_lthree0_core3_thread0_inst2_n6_aliasMSR: 2006_8A00h
	Init: _ccd0_lthree0_core3_thread1_inst2_n7_aliasMSR: 2006_8B00h
	Init: _ccd0_lthree0_core4_thread0_inst2_n8_aliasMSR: 2008_8A00h
	Init: _ccd0_lthree0_core4_thread1_inst2_n9_aliasMSR: 2008_8B00h
	Init: _ccd0_lthree0_core5_thread0_inst2_n10_aliasMSR: 200A_8A00h
	Init: _ccd0_lthree0_core5_thread1_inst2_n11_aliasMSR: 200A_8B00h
	Init: _ccd0_lthree0_core6_thread0_inst2_n12_aliasMSR: 200C_8A00h
	Init: _ccd0_lthree0_core6_thread1_inst2_n13_aliasMSR: 200C_8B00h
	Init: _ccd0_lthree0_core7_thread0_inst2_n14_aliasMSR: 200E_8A00h
	Init: _ccd0_lthree0_core7_thread1_inst2_n15_aliasMSR: 200E_8B00h
	Init: _ccd0_lthree0_core8_thread0_inst2_n16_aliasMSR: 2010_8A00h
	Init: _ccd0_lthree0_core8_thread1_inst2_n17_aliasMSR: 2010_8B00h
	Init: _ccd0_lthree0_core9_thread0_inst2_n18_aliasMSR: 2012_8A00h
	Init: _ccd0_lthree0_core9_thread1_inst2_n19_aliasMSR: 2012_8B00h
	Init: _ccd0_lthree0_core10_thread0_inst2_n20_aliasMSR: 2014_8A00h
	Init: _ccd0_lthree0_core10_thread1_inst2_n21_aliasMSR: 2014_8B00h
	Init: _ccd0_lthree0_core11_thread0_inst2_n22_aliasMSR: 2016_8A00h
	Init: _ccd0_lthree0_core11_thread1_inst2_n23_aliasMSR: 2016_8B00h
	Init: _ccd0_lthree0_core12_thread0_inst2_n24_aliasMSR: 2018_8A00h
	Init: _ccd0_lthree0_core12_thread1_inst2_n25_aliasMSR: 2018_8B00h
	Init: _ccd0_lthree0_core13_thread0_inst2_n26_aliasMSR: 201A_8A00h
	Init: _ccd0_lthree0_core13_thread1_inst2_n27_aliasMSR: 201A_8B00h
	Init: _ccd0_lthree0_core14_thread0_inst2_n28_aliasMSR: 201C_8A00h
	Init: _ccd0_lthree0_core14_thread1_inst2_n29_aliasMSR: 201C_8B00h
	Init: _ccd0_lthree0_core15_thread0_inst2_n30_aliasMSR: 201E_8A00h
	Init: _ccd0_lthree0_core15_thread1_inst2_n31_aliasMSR: 201E_8B00h
	Init: _ccd1_lthree0_core0_thread0_inst2_n32_aliasMSR: 2080_8A00h
	Init: _ccd1_lthree0_core0_thread1_inst2_n33_aliasMSR: 2080_8B00h
	Init: _ccd1_lthree0_core1_thread0_inst2_n34_aliasMSR: 2082_8A00h
	Init: _ccd1_lthree0_core1_thread1_inst2_n35_aliasMSR: 2082_8B00h
	Init: _ccd1_lthree0_core2_thread0_inst2_n36_aliasMSR: 2084_8A00h
	Init: _ccd1_lthree0_core2_thread1_inst2_n37_aliasMSR: 2084_8B00h
	Init: _ccd1_lthree0_core3_thread0_inst2_n38_aliasMSR: 2086_8A00h
	Init: _ccd1_lthree0_core3_thread1_inst2_n39_aliasMSR: 2086_8B00h
	Init: _ccd1_lthree0_core4_thread0_inst2_n40_aliasMSR: 2088_8A00h
	Init: _ccd1_lthree0_core4_thread1_inst2_n41_aliasMSR: 2088_8B00h
	Init: _ccd1_lthree0_core5_thread0_inst2_n42_aliasMSR: 208A_8A00h
	Init: _ccd1_lthree0_core5_thread1_inst2_n43_aliasMSR: 208A_8B00h
	Init: _ccd1_lthree0_core6_thread0_inst2_n44_aliasMSR: 208C_8A00h
	Init: _ccd1_lthree0_core6_thread1_inst2_n45_aliasMSR: 208C_8B00h
	Init: _ccd1_lthree0_core7_thread0_inst2_n46_aliasMSR: 208E_8A00h
	Init: _ccd1_lthree0_core7_thread1_inst2_n47_aliasMSR: 208E_8B00h
	Init: _ccd1_lthree0_core8_thread0_inst2_n48_aliasMSR: 2090_8A00h

Init: _ccd1_lthree0_core8_thread1_inst2_n49_aliasMSR: 2090_8B00h
Init: _ccd1_lthree0_core9_thread0_inst2_n50_aliasMSR: 2092_8A00h
Init: _ccd1_lthree0_core9_thread1_inst2_n51_aliasMSR: 2092_8B00h
Init: _ccd1_lthree0_core10_thread0_inst2_n52_aliasMSR: 2094_8A00h
Init: _ccd1_lthree0_core10_thread1_inst2_n53_aliasMSR: 2094_8B00h
Init: _ccd1_lthree0_core11_thread0_inst2_n54_aliasMSR: 2096_8A00h
Init: _ccd1_lthree0_core11_thread1_inst2_n55_aliasMSR: 2096_8B00h
Init: _ccd1_lthree0_core12_thread0_inst2_n56_aliasMSR: 2098_8A00h
Init: _ccd1_lthree0_core12_thread1_inst2_n57_aliasMSR: 2098_8B00h
Init: _ccd1_lthree0_core13_thread0_inst2_n58_aliasMSR: 209A_8A00h
Init: _ccd1_lthree0_core13_thread1_inst2_n59_aliasMSR: 209A_8B00h
Init: _ccd1_lthree0_core14_thread0_inst2_n60_aliasMSR: 209C_8A00h
Init: _ccd1_lthree0_core14_thread1_inst2_n61_aliasMSR: 209C_8B00h
Init: _ccd1_lthree0_core15_thread0_inst2_n62_aliasMSR: 209E_8A00h
Init: _ccd1_lthree0_core15_thread1_inst2_n63_aliasMSR: 209E_8B00h
Init: _ccd2_lthree0_core0_thread0_inst2_n64_aliasMSR: 2100_8A00h
Init: _ccd2_lthree0_core0_thread1_inst2_n65_aliasMSR: 2100_8B00h
Init: _ccd2_lthree0_core1_thread0_inst2_n66_aliasMSR: 2102_8A00h
Init: _ccd2_lthree0_core1_thread1_inst2_n67_aliasMSR: 2102_8B00h
Init: _ccd2_lthree0_core2_thread0_inst2_n68_aliasMSR: 2104_8A00h
Init: _ccd2_lthree0_core2_thread1_inst2_n69_aliasMSR: 2104_8B00h
Init: _ccd2_lthree0_core3_thread0_inst2_n70_aliasMSR: 2106_8A00h
Init: _ccd2_lthree0_core3_thread1_inst2_n71_aliasMSR: 2106_8B00h
Init: _ccd2_lthree0_core4_thread0_inst2_n72_aliasMSR: 2108_8A00h
Init: _ccd2_lthree0_core4_thread1_inst2_n73_aliasMSR: 2108_8B00h
Init: _ccd2_lthree0_core5_thread0_inst2_n74_aliasMSR: 210A_8A00h
Init: _ccd2_lthree0_core5_thread1_inst2_n75_aliasMSR: 210A_8B00h
Init: _ccd2_lthree0_core6_thread0_inst2_n76_aliasMSR: 210C_8A00h
Init: _ccd2_lthree0_core6_thread1_inst2_n77_aliasMSR: 210C_8B00h
Init: _ccd2_lthree0_core7_thread0_inst2_n78_aliasMSR: 210E_8A00h
Init: _ccd2_lthree0_core7_thread1_inst2_n79_aliasMSR: 210E_8B00h
Init: _ccd2_lthree0_core8_thread0_inst2_n80_aliasMSR: 2110_8A00h
Init: _ccd2_lthree0_core8_thread1_inst2_n81_aliasMSR: 2110_8B00h
Init: _ccd2_lthree0_core9_thread0_inst2_n82_aliasMSR: 2112_8A00h
Init: _ccd2_lthree0_core9_thread1_inst2_n83_aliasMSR: 2112_8B00h
Init: _ccd2_lthree0_core10_thread0_inst2_n84_aliasMSR: 2114_8A00h
Init: _ccd2_lthree0_core10_thread1_inst2_n85_aliasMSR: 2114_8B00h
Init: _ccd2_lthree0_core11_thread0_inst2_n86_aliasMSR: 2116_8A00h
Init: _ccd2_lthree0_core11_thread1_inst2_n87_aliasMSR: 2116_8B00h
Init: _ccd2_lthree0_core12_thread0_inst2_n88_aliasMSR: 2118_8A00h
Init: _ccd2_lthree0_core12_thread1_inst2_n89_aliasMSR: 2118_8B00h
Init: _ccd2_lthree0_core13_thread0_inst2_n90_aliasMSR: 211A_8A00h
Init: _ccd2_lthree0_core13_thread1_inst2_n91_aliasMSR: 211A_8B00h
Init: _ccd2_lthree0_core14_thread0_inst2_n92_aliasMSR: 211C_8A00h
Init: _ccd2_lthree0_core14_thread1_inst2_n93_aliasMSR: 211C_8B00h
Init: _ccd2_lthree0_core15_thread0_inst2_n94_aliasMSR: 211E_8A00h
Init: _ccd2_lthree0_core15_thread1_inst2_n95_aliasMSR: 211E_8B00h
Init: _ccd3_lthree0_core0_thread0_inst2_n96_aliasMSR: 2180_8A00h
Init: _ccd3_lthree0_core0_thread1_inst2_n97_aliasMSR: 2180_8B00h
Init: _ccd3_lthree0_core1_thread0_inst2_n98_aliasMSR: 2182_8A00h
Init: _ccd3_lthree0_core1_thread1_inst2_n99_aliasMSR: 2182_8B00h
Init: _ccd3_lthree0_core2_thread0_inst2_n100_aliasMSR: 2184_8A00h
Init: _ccd3_lthree0_core2_thread1_inst2_n101_aliasMSR: 2184_8B00h
Init: _ccd3_lthree0_core3_thread0_inst2_n102_aliasMSR: 2186_8A00h
Init: _ccd3_lthree0_core3_thread1_inst2_n103_aliasMSR: 2186_8B00h
Init: _ccd3_lthree0_core4_thread0_inst2_n104_aliasMSR: 2188_8A00h
Init: _ccd3_lthree0_core4_thread1_inst2_n105_aliasMSR: 2188_8B00h
Init: _ccd3_lthree0_core5_thread0_inst2_n106_aliasMSR: 218A_8A00h
Init: _ccd3_lthree0_core5_thread1_inst2_n107_aliasMSR: 218A_8B00h
Init: _ccd3_lthree0_core6_thread0_inst2_n108_aliasMSR: 218C_8A00h
Init: _ccd3_lthree0_core6_thread1_inst2_n109_aliasMSR: 218C_8B00h
Init: _ccd3_lthree0_core7_thread0_inst2_n110_aliasMSR: 218E_8A00h
Init: _ccd3_lthree0_core7_thread1_inst2_n111_aliasMSR: 218E_8B00h
Init: _ccd3_lthree0_core8_thread0_inst2_n112_aliasMSR: 2190_8A00h
Init: _ccd3_lthree0_core8_thread1_inst2_n113_aliasMSR: 2190_8B00h

Init: _ccd3_lthree0_core9_thread0_inst2_n114_aliasMSR: 2192_8A00h
Init: _ccd3_lthree0_core9_thread1_inst2_n115_aliasMSR: 2192_8B00h
Init: _ccd3_lthree0_core10_thread0_inst2_n116_aliasMSR: 2194_8A00h
Init: _ccd3_lthree0_core10_thread1_inst2_n117_aliasMSR: 2194_8B00h
Init: _ccd3_lthree0_core11_thread0_inst2_n118_aliasMSR: 2196_8A00h
Init: _ccd3_lthree0_core11_thread1_inst2_n119_aliasMSR: 2196_8B00h
Init: _ccd3_lthree0_core12_thread0_inst2_n120_aliasMSR: 2198_8A00h
Init: _ccd3_lthree0_core12_thread1_inst2_n121_aliasMSR: 2198_8B00h
Init: _ccd3_lthree0_core13_thread0_inst2_n122_aliasMSR: 219A_8A00h
Init: _ccd3_lthree0_core13_thread1_inst2_n123_aliasMSR: 219A_8B00h
Init: _ccd3_lthree0_core14_thread0_inst2_n124_aliasMSR: 219C_8A00h
Init: _ccd3_lthree0_core14_thread1_inst2_n125_aliasMSR: 219C_8B00h
Init: _ccd3_lthree0_core15_thread0_inst2_n126_aliasMSR: 219E_8A00h
Init: _ccd3_lthree0_core15_thread1_inst2_n127_aliasMSR: 219E_8B00h
Init: _ccd4_lthree0_core0_thread0_inst2_n128_aliasMSR: 2200_8A00h
Init: _ccd4_lthree0_core0_thread1_inst2_n129_aliasMSR: 2200_8B00h
Init: _ccd4_lthree0_core1_thread0_inst2_n130_aliasMSR: 2202_8A00h
Init: _ccd4_lthree0_core1_thread1_inst2_n131_aliasMSR: 2202_8B00h
Init: _ccd4_lthree0_core2_thread0_inst2_n132_aliasMSR: 2204_8A00h
Init: _ccd4_lthree0_core2_thread1_inst2_n133_aliasMSR: 2204_8B00h
Init: _ccd4_lthree0_core3_thread0_inst2_n134_aliasMSR: 2206_8A00h
Init: _ccd4_lthree0_core3_thread1_inst2_n135_aliasMSR: 2206_8B00h
Init: _ccd4_lthree0_core4_thread0_inst2_n136_aliasMSR: 2208_8A00h
Init: _ccd4_lthree0_core4_thread1_inst2_n137_aliasMSR: 2208_8B00h
Init: _ccd4_lthree0_core5_thread0_inst2_n138_aliasMSR: 220A_8A00h
Init: _ccd4_lthree0_core5_thread1_inst2_n139_aliasMSR: 220A_8B00h
Init: _ccd4_lthree0_core6_thread0_inst2_n140_aliasMSR: 220C_8A00h
Init: _ccd4_lthree0_core6_thread1_inst2_n141_aliasMSR: 220C_8B00h
Init: _ccd4_lthree0_core7_thread0_inst2_n142_aliasMSR: 220E_8A00h
Init: _ccd4_lthree0_core7_thread1_inst2_n143_aliasMSR: 220E_8B00h
Init: _ccd4_lthree0_core8_thread0_inst2_n144_aliasMSR: 2210_8A00h
Init: _ccd4_lthree0_core8_thread1_inst2_n145_aliasMSR: 2210_8B00h
Init: _ccd4_lthree0_core9_thread0_inst2_n146_aliasMSR: 2212_8A00h
Init: _ccd4_lthree0_core9_thread1_inst2_n147_aliasMSR: 2212_8B00h
Init: _ccd4_lthree0_core10_thread0_inst2_n148_aliasMSR: 2214_8A00h
Init: _ccd4_lthree0_core10_thread1_inst2_n149_aliasMSR: 2214_8B00h
Init: _ccd4_lthree0_core11_thread0_inst2_n150_aliasMSR: 2216_8A00h
Init: _ccd4_lthree0_core11_thread1_inst2_n151_aliasMSR: 2216_8B00h
Init: _ccd4_lthree0_core12_thread0_inst2_n152_aliasMSR: 2218_8A00h
Init: _ccd4_lthree0_core12_thread1_inst2_n153_aliasMSR: 2218_8B00h
Init: _ccd4_lthree0_core13_thread0_inst2_n154_aliasMSR: 221A_8A00h
Init: _ccd4_lthree0_core13_thread1_inst2_n155_aliasMSR: 221A_8B00h
Init: _ccd4_lthree0_core14_thread0_inst2_n156_aliasMSR: 221C_8A00h
Init: _ccd4_lthree0_core14_thread1_inst2_n157_aliasMSR: 221C_8B00h
Init: _ccd4_lthree0_core15_thread0_inst2_n158_aliasMSR: 221E_8A00h
Init: _ccd4_lthree0_core15_thread1_inst2_n159_aliasMSR: 221E_8B00h
Init: _ccd5_lthree0_core0_thread0_inst2_n160_aliasMSR: 2280_8A00h
Init: _ccd5_lthree0_core0_thread1_inst2_n161_aliasMSR: 2280_8B00h
Init: _ccd5_lthree0_core1_thread0_inst2_n162_aliasMSR: 2282_8A00h
Init: _ccd5_lthree0_core1_thread1_inst2_n163_aliasMSR: 2282_8B00h
Init: _ccd5_lthree0_core2_thread0_inst2_n164_aliasMSR: 2284_8A00h
Init: _ccd5_lthree0_core2_thread1_inst2_n165_aliasMSR: 2284_8B00h
Init: _ccd5_lthree0_core3_thread0_inst2_n166_aliasMSR: 2286_8A00h
Init: _ccd5_lthree0_core3_thread1_inst2_n167_aliasMSR: 2286_8B00h
Init: _ccd5_lthree0_core4_thread0_inst2_n168_aliasMSR: 2288_8A00h
Init: _ccd5_lthree0_core4_thread1_inst2_n169_aliasMSR: 2288_8B00h
Init: _ccd5_lthree0_core5_thread0_inst2_n170_aliasMSR: 228A_8A00h
Init: _ccd5_lthree0_core5_thread1_inst2_n171_aliasMSR: 228A_8B00h
Init: _ccd5_lthree0_core6_thread0_inst2_n172_aliasMSR: 228C_8A00h
Init: _ccd5_lthree0_core6_thread1_inst2_n173_aliasMSR: 228C_8B00h
Init: _ccd5_lthree0_core7_thread0_inst2_n174_aliasMSR: 228E_8A00h
Init: _ccd5_lthree0_core7_thread1_inst2_n175_aliasMSR: 228E_8B00h
Init: _ccd5_lthree0_core8_thread0_inst2_n176_aliasMSR: 2290_8A00h
Init: _ccd5_lthree0_core8_thread1_inst2_n177_aliasMSR: 2290_8B00h
Init: _ccd5_lthree0_core9_thread0_inst2_n178_aliasMSR: 2292_8A00h

Init: _ccd5_lthree0_core9_thread1_inst2_n179_aliasMSR: 2292_8B00h
Init: _ccd5_lthree0_core10_thread0_inst2_n180_aliasMSR: 2294_8A00h
Init: _ccd5_lthree0_core10_thread1_inst2_n181_aliasMSR: 2294_8B00h
Init: _ccd5_lthree0_core11_thread0_inst2_n182_aliasMSR: 2296_8A00h
Init: _ccd5_lthree0_core11_thread1_inst2_n183_aliasMSR: 2296_8B00h
Init: _ccd5_lthree0_core12_thread0_inst2_n184_aliasMSR: 2298_8A00h
Init: _ccd5_lthree0_core12_thread1_inst2_n185_aliasMSR: 2298_8B00h
Init: _ccd5_lthree0_core13_thread0_inst2_n186_aliasMSR: 229A_8A00h
Init: _ccd5_lthree0_core13_thread1_inst2_n187_aliasMSR: 229A_8B00h
Init: _ccd5_lthree0_core14_thread0_inst2_n188_aliasMSR: 229C_8A00h
Init: _ccd5_lthree0_core14_thread1_inst2_n189_aliasMSR: 229C_8B00h
Init: _ccd5_lthree0_core15_thread0_inst2_n190_aliasMSR: 229E_8A00h
Init: _ccd5_lthree0_core15_thread1_inst2_n191_aliasMSR: 229E_8B00h
Init: _ccd6_lthree0_core0_thread0_inst2_n192_aliasMSR: 2300_8A00h
Init: _ccd6_lthree0_core0_thread1_inst2_n193_aliasMSR: 2300_8B00h
Init: _ccd6_lthree0_core1_thread0_inst2_n194_aliasMSR: 2302_8A00h
Init: _ccd6_lthree0_core1_thread1_inst2_n195_aliasMSR: 2302_8B00h
Init: _ccd6_lthree0_core2_thread0_inst2_n196_aliasMSR: 2304_8A00h
Init: _ccd6_lthree0_core2_thread1_inst2_n197_aliasMSR: 2304_8B00h
Init: _ccd6_lthree0_core3_thread0_inst2_n198_aliasMSR: 2306_8A00h
Init: _ccd6_lthree0_core3_thread1_inst2_n199_aliasMSR: 2306_8B00h
Init: _ccd6_lthree0_core4_thread0_inst2_n200_aliasMSR: 2308_8A00h
Init: _ccd6_lthree0_core4_thread1_inst2_n201_aliasMSR: 2308_8B00h
Init: _ccd6_lthree0_core5_thread0_inst2_n202_aliasMSR: 230A_8A00h
Init: _ccd6_lthree0_core5_thread1_inst2_n203_aliasMSR: 230A_8B00h
Init: _ccd6_lthree0_core6_thread0_inst2_n204_aliasMSR: 230C_8A00h
Init: _ccd6_lthree0_core6_thread1_inst2_n205_aliasMSR: 230C_8B00h
Init: _ccd6_lthree0_core7_thread0_inst2_n206_aliasMSR: 230E_8A00h
Init: _ccd6_lthree0_core7_thread1_inst2_n207_aliasMSR: 230E_8B00h
Init: _ccd6_lthree0_core8_thread0_inst2_n208_aliasMSR: 2310_8A00h
Init: _ccd6_lthree0_core8_thread1_inst2_n209_aliasMSR: 2310_8B00h
Init: _ccd6_lthree0_core9_thread0_inst2_n210_aliasMSR: 2312_8A00h
Init: _ccd6_lthree0_core9_thread1_inst2_n211_aliasMSR: 2312_8B00h
Init: _ccd6_lthree0_core10_thread0_inst2_n212_aliasMSR: 2314_8A00h
Init: _ccd6_lthree0_core10_thread1_inst2_n213_aliasMSR: 2314_8B00h
Init: _ccd6_lthree0_core11_thread0_inst2_n214_aliasMSR: 2316_8A00h
Init: _ccd6_lthree0_core11_thread1_inst2_n215_aliasMSR: 2316_8B00h
Init: _ccd6_lthree0_core12_thread0_inst2_n216_aliasMSR: 2318_8A00h
Init: _ccd6_lthree0_core12_thread1_inst2_n217_aliasMSR: 2318_8B00h
Init: _ccd6_lthree0_core13_thread0_inst2_n218_aliasMSR: 231A_8A00h
Init: _ccd6_lthree0_core13_thread1_inst2_n219_aliasMSR: 231A_8B00h
Init: _ccd6_lthree0_core14_thread0_inst2_n220_aliasMSR: 231C_8A00h
Init: _ccd6_lthree0_core14_thread1_inst2_n221_aliasMSR: 231C_8B00h
Init: _ccd6_lthree0_core15_thread0_inst2_n222_aliasMSR: 231E_8A00h
Init: _ccd6_lthree0_core15_thread1_inst2_n223_aliasMSR: 231E_8B00h
Init: _ccd7_lthree0_core0_thread0_inst2_n224_aliasMSR: 2380_8A00h
Init: _ccd7_lthree0_core0_thread1_inst2_n225_aliasMSR: 2380_8B00h
Init: _ccd7_lthree0_core1_thread0_inst2_n226_aliasMSR: 2382_8A00h
Init: _ccd7_lthree0_core1_thread1_inst2_n227_aliasMSR: 2382_8B00h
Init: _ccd7_lthree0_core2_thread0_inst2_n228_aliasMSR: 2384_8A00h
Init: _ccd7_lthree0_core2_thread1_inst2_n229_aliasMSR: 2384_8B00h
Init: _ccd7_lthree0_core3_thread0_inst2_n230_aliasMSR: 2386_8A00h
Init: _ccd7_lthree0_core3_thread1_inst2_n231_aliasMSR: 2386_8B00h
Init: _ccd7_lthree0_core4_thread0_inst2_n232_aliasMSR: 2388_8A00h
Init: _ccd7_lthree0_core4_thread1_inst2_n233_aliasMSR: 2388_8B00h
Init: _ccd7_lthree0_core5_thread0_inst2_n234_aliasMSR: 238A_8A00h
Init: _ccd7_lthree0_core5_thread1_inst2_n235_aliasMSR: 238A_8B00h
Init: _ccd7_lthree0_core6_thread0_inst2_n236_aliasMSR: 238C_8A00h
Init: _ccd7_lthree0_core6_thread1_inst2_n237_aliasMSR: 238C_8B00h
Init: _ccd7_lthree0_core7_thread0_inst2_n238_aliasMSR: 238E_8A00h
Init: _ccd7_lthree0_core7_thread1_inst2_n239_aliasMSR: 238E_8B00h
Init: _ccd7_lthree0_core8_thread0_inst2_n240_aliasMSR: 2390_8A00h
Init: _ccd7_lthree0_core8_thread1_inst2_n241_aliasMSR: 2390_8B00h
Init: _ccd7_lthree0_core9_thread0_inst2_n242_aliasMSR: 2392_8A00h
Init: _ccd7_lthree0_core9_thread1_inst2_n243_aliasMSR: 2392_8B00h

Init: _ccd7_lthree0_core10_thread0_inst2_n244_aliasMSR: 2394_8A00h
Init: _ccd7_lthree0_core10_thread1_inst2_n245_aliasMSR: 2394_8B00h
Init: _ccd7_lthree0_core11_thread0_inst2_n246_aliasMSR: 2396_8A00h
Init: _ccd7_lthree0_core11_thread1_inst2_n247_aliasMSR: 2396_8B00h
Init: _ccd7_lthree0_core12_thread0_inst2_n248_aliasMSR: 2398_8A00h
Init: _ccd7_lthree0_core12_thread1_inst2_n249_aliasMSR: 2398_8B00h
Init: _ccd7_lthree0_core13_thread0_inst2_n250_aliasMSR: 239A_8A00h
Init: _ccd7_lthree0_core13_thread1_inst2_n251_aliasMSR: 239A_8B00h
Init: _ccd7_lthree0_core14_thread0_inst2_n252_aliasMSR: 239C_8A00h
Init: _ccd7_lthree0_core14_thread1_inst2_n253_aliasMSR: 239C_8B00h
Init: _ccd7_lthree0_core15_thread0_inst2_n254_aliasMSR: 239E_8A00h
Init: _ccd7_lthree0_core15_thread1_inst2_n255_aliasMSR: 239E_8B00h
Init: _ccd8_lthree0_core0_thread0_inst2_n256_aliasMSR: 2400_8A00h
Init: _ccd8_lthree0_core0_thread1_inst2_n257_aliasMSR: 2400_8B00h
Init: _ccd8_lthree0_core1_thread0_inst2_n258_aliasMSR: 2402_8A00h
Init: _ccd8_lthree0_core1_thread1_inst2_n259_aliasMSR: 2402_8B00h
Init: _ccd8_lthree0_core2_thread0_inst2_n260_aliasMSR: 2404_8A00h
Init: _ccd8_lthree0_core2_thread1_inst2_n261_aliasMSR: 2404_8B00h
Init: _ccd8_lthree0_core3_thread0_inst2_n262_aliasMSR: 2406_8A00h
Init: _ccd8_lthree0_core3_thread1_inst2_n263_aliasMSR: 2406_8B00h
Init: _ccd8_lthree0_core4_thread0_inst2_n264_aliasMSR: 2408_8A00h
Init: _ccd8_lthree0_core4_thread1_inst2_n265_aliasMSR: 2408_8B00h
Init: _ccd8_lthree0_core5_thread0_inst2_n266_aliasMSR: 240A_8A00h
Init: _ccd8_lthree0_core5_thread1_inst2_n267_aliasMSR: 240A_8B00h
Init: _ccd8_lthree0_core6_thread0_inst2_n268_aliasMSR: 240C_8A00h
Init: _ccd8_lthree0_core6_thread1_inst2_n269_aliasMSR: 240C_8B00h
Init: _ccd8_lthree0_core7_thread0_inst2_n270_aliasMSR: 240E_8A00h
Init: _ccd8_lthree0_core7_thread1_inst2_n271_aliasMSR: 240E_8B00h
Init: _ccd8_lthree0_core8_thread0_inst2_n272_aliasMSR: 2410_8A00h
Init: _ccd8_lthree0_core8_thread1_inst2_n273_aliasMSR: 2410_8B00h
Init: _ccd8_lthree0_core9_thread0_inst2_n274_aliasMSR: 2412_8A00h
Init: _ccd8_lthree0_core9_thread1_inst2_n275_aliasMSR: 2412_8B00h
Init: _ccd8_lthree0_core10_thread0_inst2_n276_aliasMSR: 2414_8A00h
Init: _ccd8_lthree0_core10_thread1_inst2_n277_aliasMSR: 2414_8B00h
Init: _ccd8_lthree0_core11_thread0_inst2_n278_aliasMSR: 2416_8A00h
Init: _ccd8_lthree0_core11_thread1_inst2_n279_aliasMSR: 2416_8B00h
Init: _ccd8_lthree0_core12_thread0_inst2_n280_aliasMSR: 2418_8A00h
Init: _ccd8_lthree0_core12_thread1_inst2_n281_aliasMSR: 2418_8B00h
Init: _ccd8_lthree0_core13_thread0_inst2_n282_aliasMSR: 241A_8A00h
Init: _ccd8_lthree0_core13_thread1_inst2_n283_aliasMSR: 241A_8B00h
Init: _ccd8_lthree0_core14_thread0_inst2_n284_aliasMSR: 241C_8A00h
Init: _ccd8_lthree0_core14_thread1_inst2_n285_aliasMSR: 241C_8B00h
Init: _ccd8_lthree0_core15_thread0_inst2_n286_aliasMSR: 241E_8A00h
Init: _ccd8_lthree0_core15_thread1_inst2_n287_aliasMSR: 241E_8B00h
Init: _ccd9_lthree0_core0_thread0_inst2_n288_aliasMSR: 2480_8A00h
Init: _ccd9_lthree0_core0_thread1_inst2_n289_aliasMSR: 2480_8B00h
Init: _ccd9_lthree0_core1_thread0_inst2_n290_aliasMSR: 2482_8A00h
Init: _ccd9_lthree0_core1_thread1_inst2_n291_aliasMSR: 2482_8B00h
Init: _ccd9_lthree0_core2_thread0_inst2_n292_aliasMSR: 2484_8A00h
Init: _ccd9_lthree0_core2_thread1_inst2_n293_aliasMSR: 2484_8B00h
Init: _ccd9_lthree0_core3_thread0_inst2_n294_aliasMSR: 2486_8A00h
Init: _ccd9_lthree0_core3_thread1_inst2_n295_aliasMSR: 2486_8B00h
Init: _ccd9_lthree0_core4_thread0_inst2_n296_aliasMSR: 2488_8A00h
Init: _ccd9_lthree0_core4_thread1_inst2_n297_aliasMSR: 2488_8B00h
Init: _ccd9_lthree0_core5_thread0_inst2_n298_aliasMSR: 248A_8A00h
Init: _ccd9_lthree0_core5_thread1_inst2_n299_aliasMSR: 248A_8B00h
Init: _ccd9_lthree0_core6_thread0_inst2_n300_aliasMSR: 248C_8A00h
Init: _ccd9_lthree0_core6_thread1_inst2_n301_aliasMSR: 248C_8B00h
Init: _ccd9_lthree0_core7_thread0_inst2_n302_aliasMSR: 248E_8A00h
Init: _ccd9_lthree0_core7_thread1_inst2_n303_aliasMSR: 248E_8B00h
Init: _ccd9_lthree0_core8_thread0_inst2_n304_aliasMSR: 2490_8A00h
Init: _ccd9_lthree0_core8_thread1_inst2_n305_aliasMSR: 2490_8B00h
Init: _ccd9_lthree0_core9_thread0_inst2_n306_aliasMSR: 2492_8A00h
Init: _ccd9_lthree0_core9_thread1_inst2_n307_aliasMSR: 2492_8B00h
Init: _ccd9_lthree0_core10_thread0_inst2_n308_aliasMSR: 2494_8A00h

Init: _ccd9_lthree0_core10_thread1_inst2_n309_aliasMSR: 2494_8B00h
Init: _ccd9_lthree0_core11_thread0_inst2_n310_aliasMSR: 2496_8A00h
Init: _ccd9_lthree0_core11_thread1_inst2_n311_aliasMSR: 2496_8B00h
Init: _ccd9_lthree0_core12_thread0_inst2_n312_aliasMSR: 2498_8A00h
Init: _ccd9_lthree0_core12_thread1_inst2_n313_aliasMSR: 2498_8B00h
Init: _ccd9_lthree0_core13_thread0_inst2_n314_aliasMSR: 249A_8A00h
Init: _ccd9_lthree0_core13_thread1_inst2_n315_aliasMSR: 249A_8B00h
Init: _ccd9_lthree0_core14_thread0_inst2_n316_aliasMSR: 249C_8A00h
Init: _ccd9_lthree0_core14_thread1_inst2_n317_aliasMSR: 249C_8B00h
Init: _ccd9_lthree0_core15_thread0_inst2_n318_aliasMSR: 249E_8A00h
Init: _ccd9_lthree0_core15_thread1_inst2_n319_aliasMSR: 249E_8B00h
Init: _ccd10_lthree0_core0_thread0_inst2_n320_aliasMSR: 2500_8A00h
Init: _ccd10_lthree0_core0_thread1_inst2_n321_aliasMSR: 2500_8B00h
Init: _ccd10_lthree0_core1_thread0_inst2_n322_aliasMSR: 2502_8A00h
Init: _ccd10_lthree0_core1_thread1_inst2_n323_aliasMSR: 2502_8B00h
Init: _ccd10_lthree0_core2_thread0_inst2_n324_aliasMSR: 2504_8A00h
Init: _ccd10_lthree0_core2_thread1_inst2_n325_aliasMSR: 2504_8B00h
Init: _ccd10_lthree0_core3_thread0_inst2_n326_aliasMSR: 2506_8A00h
Init: _ccd10_lthree0_core3_thread1_inst2_n327_aliasMSR: 2506_8B00h
Init: _ccd10_lthree0_core4_thread0_inst2_n328_aliasMSR: 2508_8A00h
Init: _ccd10_lthree0_core4_thread1_inst2_n329_aliasMSR: 2508_8B00h
Init: _ccd10_lthree0_core5_thread0_inst2_n330_aliasMSR: 250A_8A00h
Init: _ccd10_lthree0_core5_thread1_inst2_n331_aliasMSR: 250A_8B00h
Init: _ccd10_lthree0_core6_thread0_inst2_n332_aliasMSR: 250C_8A00h
Init: _ccd10_lthree0_core6_thread1_inst2_n333_aliasMSR: 250C_8B00h
Init: _ccd10_lthree0_core7_thread0_inst2_n334_aliasMSR: 250E_8A00h
Init: _ccd10_lthree0_core7_thread1_inst2_n335_aliasMSR: 250E_8B00h
Init: _ccd10_lthree0_core8_thread0_inst2_n336_aliasMSR: 2510_8A00h
Init: _ccd10_lthree0_core8_thread1_inst2_n337_aliasMSR: 2510_8B00h
Init: _ccd10_lthree0_core9_thread0_inst2_n338_aliasMSR: 2512_8A00h
Init: _ccd10_lthree0_core9_thread1_inst2_n339_aliasMSR: 2512_8B00h
Init: _ccd10_lthree0_core10_thread0_inst2_n340_aliasMSR: 2514_8A00h
Init: _ccd10_lthree0_core10_thread1_inst2_n341_aliasMSR: 2514_8B00h
Init: _ccd10_lthree0_core11_thread0_inst2_n342_aliasMSR: 2516_8A00h
Init: _ccd10_lthree0_core11_thread1_inst2_n343_aliasMSR: 2516_8B00h
Init: _ccd10_lthree0_core12_thread0_inst2_n344_aliasMSR: 2518_8A00h
Init: _ccd10_lthree0_core12_thread1_inst2_n345_aliasMSR: 2518_8B00h
Init: _ccd10_lthree0_core13_thread0_inst2_n346_aliasMSR: 251A_8A00h
Init: _ccd10_lthree0_core13_thread1_inst2_n347_aliasMSR: 251A_8B00h
Init: _ccd10_lthree0_core14_thread0_inst2_n348_aliasMSR: 251C_8A00h
Init: _ccd10_lthree0_core14_thread1_inst2_n349_aliasMSR: 251C_8B00h
Init: _ccd10_lthree0_core15_thread0_inst2_n350_aliasMSR: 251E_8A00h
Init: _ccd10_lthree0_core15_thread1_inst2_n351_aliasMSR: 251E_8B00h
Init: _ccd11_lthree0_core0_thread0_inst2_n352_aliasMSR: 2580_8A00h
Init: _ccd11_lthree0_core0_thread1_inst2_n353_aliasMSR: 2580_8B00h
Init: _ccd11_lthree0_core1_thread0_inst2_n354_aliasMSR: 2582_8A00h
Init: _ccd11_lthree0_core1_thread1_inst2_n355_aliasMSR: 2582_8B00h
Init: _ccd11_lthree0_core2_thread0_inst2_n356_aliasMSR: 2584_8A00h
Init: _ccd11_lthree0_core2_thread1_inst2_n357_aliasMSR: 2584_8B00h
Init: _ccd11_lthree0_core3_thread0_inst2_n358_aliasMSR: 2586_8A00h
Init: _ccd11_lthree0_core3_thread1_inst2_n359_aliasMSR: 2586_8B00h
Init: _ccd11_lthree0_core4_thread0_inst2_n360_aliasMSR: 2588_8A00h
Init: _ccd11_lthree0_core4_thread1_inst2_n361_aliasMSR: 2588_8B00h
Init: _ccd11_lthree0_core5_thread0_inst2_n362_aliasMSR: 258A_8A00h
Init: _ccd11_lthree0_core5_thread1_inst2_n363_aliasMSR: 258A_8B00h
Init: _ccd11_lthree0_core6_thread0_inst2_n364_aliasMSR: 258C_8A00h
Init: _ccd11_lthree0_core6_thread1_inst2_n365_aliasMSR: 258C_8B00h
Init: _ccd11_lthree0_core7_thread0_inst2_n366_aliasMSR: 258E_8A00h
Init: _ccd11_lthree0_core7_thread1_inst2_n367_aliasMSR: 258E_8B00h
Init: _ccd11_lthree0_core8_thread0_inst2_n368_aliasMSR: 2590_8A00h
Init: _ccd11_lthree0_core8_thread1_inst2_n369_aliasMSR: 2590_8B00h
Init: _ccd11_lthree0_core9_thread0_inst2_n370_aliasMSR: 2592_8A00h
Init: _ccd11_lthree0_core9_thread1_inst2_n371_aliasMSR: 2592_8B00h
Init: _ccd11_lthree0_core10_thread0_inst2_n372_aliasMSR: 2594_8A00h
Init: _ccd11_lthree0_core10_thread1_inst2_n373_aliasMSR: 2594_8B00h

Init: _ccd11_lthree0_core11_thread0_inst2_n374_aliasMSR: 2596_8A00h
Init: _ccd11_lthree0_core11_thread1_inst2_n375_aliasMSR: 2596_8B00h
Init: _ccd11_lthree0_core12_thread0_inst2_n376_aliasMSR: 2598_8A00h
Init: _ccd11_lthree0_core12_thread1_inst2_n377_aliasMSR: 2598_8B00h
Init: _ccd11_lthree0_core13_thread0_inst2_n378_aliasMSR: 259A_8A00h
Init: _ccd11_lthree0_core13_thread1_inst2_n379_aliasMSR: 259A_8B00h
Init: _ccd11_lthree0_core14_thread0_inst2_n380_aliasMSR: 259C_8A00h
Init: _ccd11_lthree0_core14_thread1_inst2_n381_aliasMSR: 259C_8B00h
Init: _ccd11_lthree0_core15_thread0_inst2_n382_aliasMSR: 259E_8A00h
Init: _ccd11_lthree0_core15_thread1_inst2_n383_aliasMSR: 259E_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](#)

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst2_n[383:0]_aliasMSR; MSRC000_2026

Bits	Description
63:32	Syndrom . 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	ErrorPriority . 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	Length . Read-write, Volatile . Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::L2::MCA_SYND_L2[Syndrome]. Length values greater than 32 (decimal) are interpreted as equal to 32 (decimal). 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	ErrorInformation . 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_L2].

Table 60: MCA_SYND_L2

Error Type	Bits	Description
MultiHit	[17:16] [15:0]	Index One-hot way vector
Tag	[17] [16:14] [13:4] [3:0]	NoError Reserved Index Way
Data	[17:16] [15] [14:5] [4] [3:0]	Reserved Poison Index TopBotLoc Way
Hwa	[17:0]	Reserved
Sdp	[17:17] [16:7] [6:4] [3:0]	Reserved Addr[15:6] 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst2_n[383:0]_aliasMSR; MSRC000_2028

Bits	Description
63	Val. Read-write, Volatile . Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	Overflow. 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 overwrite priorities.)
61:59	RESERV4. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	AddrV. Read-write, Volatile . Reset: Cold,0. 1=MCA::L2::MCA_DEADDR_L2 contains address information associated with the error.
57:54	RESERV3. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	SyndV. 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	RESERV2. Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	Deferred. 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	RESERV1. Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	AddrLsb. 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	RESERV0. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	ErrorCodeExt. 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	ErrorCode. 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].

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst2_n[383:0]_aliasMSR; MSRC000_2029

Bits	Description
63:0	ErrorAddr. 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.

`_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst2_n[383:0]_aliasMSR; MSRC001_0402`

Bits	Description
63:7	Reserved.
6	Wdt. Read-write. Reset: 0. Reserved
5	StateMachine. Read-write. Reset: 0. Error initiated by programmable state machine.
4	Sdp. Read-write. Reset: 0. SDP Read Response Parity Error
3	Hwa. Read-write. Reset: 0. Hardware Assert Error.
2	Data. Read-write. Reset: 0. L2M Data Array ECC Error.
1	Tag. Read-write. Reset: 0. L2M Tag or State Array ECC Error.
0	MultiHit. 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](#)`_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst2_n[383:0]_aliasMSR; MSRC000_202E`

Bits	Description
63:0	Syndrome. 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](#)`_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst2_n[383:0]_aliasMSR; MSRC000_202F`

Bits	Description
63:0	Syndrome. 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.	
_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst3_n[383:0]_aliasMSRLEGACY; MSR0000_040C	
_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst3_n[383:0]_aliasMSR; MSRC000_2030	
Bits	Description
63:10	Reserved.
9	HWA. Read-write. Reset: 0. Hardware Assertion Error.
8	OCQ. Read-write. Reset: 0. Micro-op fetch queue parity error.
7	UcSeq. Read-write. Reset: 0. Patch RAM sequencer parity error.
6	UcDat. Read-write. Reset: 0. Patch RAM data parity error.
5	Faq. Read-write. Reset: 0. Fetch address FIFO parity error.
4	Idq. Read-write. Reset: 0. Instruction dispatch queue parity error.
3	UopQ. Read-write. Reset: 0. Micro-op queue parity error.
2	Ibq. Read-write. Reset: 0. Instruction buffer parity error.
1	OcDat. Read-write. Reset: 0. Micro-op cache data parity error.
0	OcTag. 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst3_n[383:0]_aliasMSRLEGACY; MSR0000_040D

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst3_n[383:0]_aliasMSR; MSRC000_2031

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. 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	MiscV. 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	AddrV. 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	PCC. 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	ErrCoreIdVal. 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	TCC. 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	RESERV54. 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	SyndV. 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	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. 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	RESERV22. 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	ErrorCodeExt. 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	ErrorCode. 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_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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst3_n[383:0]_aliasMSRLEGACY; MSR0000_040E

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst3_n[383:0]_aliasMSR; MSRC000_2032

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::DE::MCA_STATUS_DE. For physical addresses, the most significant bit is given by Core::X86::Cpuid::LongModeInfo[PhysAddrSize].

Table 62: 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst3_n[383:0]_aliasMSRLEGACY; MSR0000_040F

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst3_n[383:0]_aliasMSR; MSRC000_2033

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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	LvtOffset. 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	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msrr::HWCR[McStatusWrEn] !MCA::DE::MCA_MISC0_DE[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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	Ovrflw. 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	ErrCnt. 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	BlkPtr. 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst3_n[383:0]_aliasMSR; MSRC000_2034

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	McaXEnable. 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	IntPresent. 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	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::DE::MCA_CONFIG_DE[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. 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	DeferredErrorLoggingSupported. 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	McaX. 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst3_n[383:0]_aliasMSR; MSRC000_2035

Bits	Description
63:48	McaType . Read-only. Reset: 0003h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi . 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	HardwareID . Read-only. Reset: 0B0h. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _ccd0_lthree0_core0_thread0_inst3_n0_aliasMSR: 2000_9600h
	Init: _ccd0_lthree0_core0_thread1_inst3_n1_aliasMSR: 2000_9700h
	Init: _ccd0_lthree0_core1_thread0_inst3_n2_aliasMSR: 2002_9600h
	Init: _ccd0_lthree0_core1_thread1_inst3_n3_aliasMSR: 2002_9700h
	Init: _ccd0_lthree0_core2_thread0_inst3_n4_aliasMSR: 2004_9600h
	Init: _ccd0_lthree0_core2_thread1_inst3_n5_aliasMSR: 2004_9700h
	Init: _ccd0_lthree0_core3_thread0_inst3_n6_aliasMSR: 2006_9600h
	Init: _ccd0_lthree0_core3_thread1_inst3_n7_aliasMSR: 2006_9700h
	Init: _ccd0_lthree0_core4_thread0_inst3_n8_aliasMSR: 2008_9600h
	Init: _ccd0_lthree0_core4_thread1_inst3_n9_aliasMSR: 2008_9700h
	Init: _ccd0_lthree0_core5_thread0_inst3_n10_aliasMSR: 200A_9600h
	Init: _ccd0_lthree0_core5_thread1_inst3_n11_aliasMSR: 200A_9700h
	Init: _ccd0_lthree0_core6_thread0_inst3_n12_aliasMSR: 200C_9600h
	Init: _ccd0_lthree0_core6_thread1_inst3_n13_aliasMSR: 200C_9700h
	Init: _ccd0_lthree0_core7_thread0_inst3_n14_aliasMSR: 200E_9600h
	Init: _ccd0_lthree0_core7_thread1_inst3_n15_aliasMSR: 200E_9700h
	Init: _ccd0_lthree0_core8_thread0_inst3_n16_aliasMSR: 2010_9600h
	Init: _ccd0_lthree0_core8_thread1_inst3_n17_aliasMSR: 2010_9700h
	Init: _ccd0_lthree0_core9_thread0_inst3_n18_aliasMSR: 2012_9600h
	Init: _ccd0_lthree0_core9_thread1_inst3_n19_aliasMSR: 2012_9700h
	Init: _ccd0_lthree0_core10_thread0_inst3_n20_aliasMSR: 2014_9600h
	Init: _ccd0_lthree0_core10_thread1_inst3_n21_aliasMSR: 2014_9700h
	Init: _ccd0_lthree0_core11_thread0_inst3_n22_aliasMSR: 2016_9600h
	Init: _ccd0_lthree0_core11_thread1_inst3_n23_aliasMSR: 2016_9700h
	Init: _ccd0_lthree0_core12_thread0_inst3_n24_aliasMSR: 2018_9600h
	Init: _ccd0_lthree0_core12_thread1_inst3_n25_aliasMSR: 2018_9700h
	Init: _ccd0_lthree0_core13_thread0_inst3_n26_aliasMSR: 201A_9600h
	Init: _ccd0_lthree0_core13_thread1_inst3_n27_aliasMSR: 201A_9700h
	Init: _ccd0_lthree0_core14_thread0_inst3_n28_aliasMSR: 201C_9600h
	Init: _ccd0_lthree0_core14_thread1_inst3_n29_aliasMSR: 201C_9700h
	Init: _ccd0_lthree0_core15_thread0_inst3_n30_aliasMSR: 201E_9600h
	Init: _ccd0_lthree0_core15_thread1_inst3_n31_aliasMSR: 201E_9700h
	Init: _ccd1_lthree0_core0_thread0_inst3_n32_aliasMSR: 2080_9600h
	Init: _ccd1_lthree0_core0_thread1_inst3_n33_aliasMSR: 2080_9700h
	Init: _ccd1_lthree0_core1_thread0_inst3_n34_aliasMSR: 2082_9600h
	Init: _ccd1_lthree0_core1_thread1_inst3_n35_aliasMSR: 2082_9700h
	Init: _ccd1_lthree0_core2_thread0_inst3_n36_aliasMSR: 2084_9600h
	Init: _ccd1_lthree0_core2_thread1_inst3_n37_aliasMSR: 2084_9700h
	Init: _ccd1_lthree0_core3_thread0_inst3_n38_aliasMSR: 2086_9600h
	Init: _ccd1_lthree0_core3_thread1_inst3_n39_aliasMSR: 2086_9700h
	Init: _ccd1_lthree0_core4_thread0_inst3_n40_aliasMSR: 2088_9600h
	Init: _ccd1_lthree0_core4_thread1_inst3_n41_aliasMSR: 2088_9700h
	Init: _ccd1_lthree0_core5_thread0_inst3_n42_aliasMSR: 208A_9600h
	Init: _ccd1_lthree0_core5_thread1_inst3_n43_aliasMSR: 208A_9700h
	Init: _ccd1_lthree0_core6_thread0_inst3_n44_aliasMSR: 208C_9600h
	Init: _ccd1_lthree0_core6_thread1_inst3_n45_aliasMSR: 208C_9700h
	Init: _ccd1_lthree0_core7_thread0_inst3_n46_aliasMSR: 208E_9600h
	Init: _ccd1_lthree0_core7_thread1_inst3_n47_aliasMSR: 208E_9700h
	Init: _ccd1_lthree0_core8_thread0_inst3_n48_aliasMSR: 2090_9600h

Init: _ccd1_lthree0_core8_thread1_inst3_n49_aliasMSR: 2090_9700h
Init: _ccd1_lthree0_core9_thread0_inst3_n50_aliasMSR: 2092_9600h
Init: _ccd1_lthree0_core9_thread1_inst3_n51_aliasMSR: 2092_9700h
Init: _ccd1_lthree0_core10_thread0_inst3_n52_aliasMSR: 2094_9600h
Init: _ccd1_lthree0_core10_thread1_inst3_n53_aliasMSR: 2094_9700h
Init: _ccd1_lthree0_core11_thread0_inst3_n54_aliasMSR: 2096_9600h
Init: _ccd1_lthree0_core11_thread1_inst3_n55_aliasMSR: 2096_9700h
Init: _ccd1_lthree0_core12_thread0_inst3_n56_aliasMSR: 2098_9600h
Init: _ccd1_lthree0_core12_thread1_inst3_n57_aliasMSR: 2098_9700h
Init: _ccd1_lthree0_core13_thread0_inst3_n58_aliasMSR: 209A_9600h
Init: _ccd1_lthree0_core13_thread1_inst3_n59_aliasMSR: 209A_9700h
Init: _ccd1_lthree0_core14_thread0_inst3_n60_aliasMSR: 209C_9600h
Init: _ccd1_lthree0_core14_thread1_inst3_n61_aliasMSR: 209C_9700h
Init: _ccd1_lthree0_core15_thread0_inst3_n62_aliasMSR: 209E_9600h
Init: _ccd1_lthree0_core15_thread1_inst3_n63_aliasMSR: 209E_9700h
Init: _ccd2_lthree0_core0_thread0_inst3_n64_aliasMSR: 2100_9600h
Init: _ccd2_lthree0_core0_thread1_inst3_n65_aliasMSR: 2100_9700h
Init: _ccd2_lthree0_core1_thread0_inst3_n66_aliasMSR: 2102_9600h
Init: _ccd2_lthree0_core1_thread1_inst3_n67_aliasMSR: 2102_9700h
Init: _ccd2_lthree0_core2_thread0_inst3_n68_aliasMSR: 2104_9600h
Init: _ccd2_lthree0_core2_thread1_inst3_n69_aliasMSR: 2104_9700h
Init: _ccd2_lthree0_core3_thread0_inst3_n70_aliasMSR: 2106_9600h
Init: _ccd2_lthree0_core3_thread1_inst3_n71_aliasMSR: 2106_9700h
Init: _ccd2_lthree0_core4_thread0_inst3_n72_aliasMSR: 2108_9600h
Init: _ccd2_lthree0_core4_thread1_inst3_n73_aliasMSR: 2108_9700h
Init: _ccd2_lthree0_core5_thread0_inst3_n74_aliasMSR: 210A_9600h
Init: _ccd2_lthree0_core5_thread1_inst3_n75_aliasMSR: 210A_9700h
Init: _ccd2_lthree0_core6_thread0_inst3_n76_aliasMSR: 210C_9600h
Init: _ccd2_lthree0_core6_thread1_inst3_n77_aliasMSR: 210C_9700h
Init: _ccd2_lthree0_core7_thread0_inst3_n78_aliasMSR: 210E_9600h
Init: _ccd2_lthree0_core7_thread1_inst3_n79_aliasMSR: 210E_9700h
Init: _ccd2_lthree0_core8_thread0_inst3_n80_aliasMSR: 2110_9600h
Init: _ccd2_lthree0_core8_thread1_inst3_n81_aliasMSR: 2110_9700h
Init: _ccd2_lthree0_core9_thread0_inst3_n82_aliasMSR: 2112_9600h
Init: _ccd2_lthree0_core9_thread1_inst3_n83_aliasMSR: 2112_9700h
Init: _ccd2_lthree0_core10_thread0_inst3_n84_aliasMSR: 2114_9600h
Init: _ccd2_lthree0_core10_thread1_inst3_n85_aliasMSR: 2114_9700h
Init: _ccd2_lthree0_core11_thread0_inst3_n86_aliasMSR: 2116_9600h
Init: _ccd2_lthree0_core11_thread1_inst3_n87_aliasMSR: 2116_9700h
Init: _ccd2_lthree0_core12_thread0_inst3_n88_aliasMSR: 2118_9600h
Init: _ccd2_lthree0_core12_thread1_inst3_n89_aliasMSR: 2118_9700h
Init: _ccd2_lthree0_core13_thread0_inst3_n90_aliasMSR: 211A_9600h
Init: _ccd2_lthree0_core13_thread1_inst3_n91_aliasMSR: 211A_9700h
Init: _ccd2_lthree0_core14_thread0_inst3_n92_aliasMSR: 211C_9600h
Init: _ccd2_lthree0_core14_thread1_inst3_n93_aliasMSR: 211C_9700h
Init: _ccd2_lthree0_core15_thread0_inst3_n94_aliasMSR: 211E_9600h
Init: _ccd2_lthree0_core15_thread1_inst3_n95_aliasMSR: 211E_9700h
Init: _ccd3_lthree0_core0_thread0_inst3_n96_aliasMSR: 2180_9600h
Init: _ccd3_lthree0_core0_thread1_inst3_n97_aliasMSR: 2180_9700h
Init: _ccd3_lthree0_core1_thread0_inst3_n98_aliasMSR: 2182_9600h
Init: _ccd3_lthree0_core1_thread1_inst3_n99_aliasMSR: 2182_9700h
Init: _ccd3_lthree0_core2_thread0_inst3_n100_aliasMSR: 2184_9600h
Init: _ccd3_lthree0_core2_thread1_inst3_n101_aliasMSR: 2184_9700h
Init: _ccd3_lthree0_core3_thread0_inst3_n102_aliasMSR: 2186_9600h
Init: _ccd3_lthree0_core3_thread1_inst3_n103_aliasMSR: 2186_9700h
Init: _ccd3_lthree0_core4_thread0_inst3_n104_aliasMSR: 2188_9600h
Init: _ccd3_lthree0_core4_thread1_inst3_n105_aliasMSR: 2188_9700h
Init: _ccd3_lthree0_core5_thread0_inst3_n106_aliasMSR: 218A_9600h
Init: _ccd3_lthree0_core5_thread1_inst3_n107_aliasMSR: 218A_9700h
Init: _ccd3_lthree0_core6_thread0_inst3_n108_aliasMSR: 218C_9600h
Init: _ccd3_lthree0_core6_thread1_inst3_n109_aliasMSR: 218C_9700h
Init: _ccd3_lthree0_core7_thread0_inst3_n110_aliasMSR: 218E_9600h
Init: _ccd3_lthree0_core7_thread1_inst3_n111_aliasMSR: 218E_9700h
Init: _ccd3_lthree0_core8_thread0_inst3_n112_aliasMSR: 2190_9600h
Init: _ccd3_lthree0_core8_thread1_inst3_n113_aliasMSR: 2190_9700h

Init: _ccd3_lthree0_core9_thread0_inst3_n114_aliasMSR: 2192_9600h
Init: _ccd3_lthree0_core9_thread1_inst3_n115_aliasMSR: 2192_9700h
Init: _ccd3_lthree0_core10_thread0_inst3_n116_aliasMSR: 2194_9600h
Init: _ccd3_lthree0_core10_thread1_inst3_n117_aliasMSR: 2194_9700h
Init: _ccd3_lthree0_core11_thread0_inst3_n118_aliasMSR: 2196_9600h
Init: _ccd3_lthree0_core11_thread1_inst3_n119_aliasMSR: 2196_9700h
Init: _ccd3_lthree0_core12_thread0_inst3_n120_aliasMSR: 2198_9600h
Init: _ccd3_lthree0_core12_thread1_inst3_n121_aliasMSR: 2198_9700h
Init: _ccd3_lthree0_core13_thread0_inst3_n122_aliasMSR: 219A_9600h
Init: _ccd3_lthree0_core13_thread1_inst3_n123_aliasMSR: 219A_9700h
Init: _ccd3_lthree0_core14_thread0_inst3_n124_aliasMSR: 219C_9600h
Init: _ccd3_lthree0_core14_thread1_inst3_n125_aliasMSR: 219C_9700h
Init: _ccd3_lthree0_core15_thread0_inst3_n126_aliasMSR: 219E_9600h
Init: _ccd3_lthree0_core15_thread1_inst3_n127_aliasMSR: 219E_9700h
Init: _ccd4_lthree0_core0_thread0_inst3_n128_aliasMSR: 2200_9600h
Init: _ccd4_lthree0_core0_thread1_inst3_n129_aliasMSR: 2200_9700h
Init: _ccd4_lthree0_core1_thread0_inst3_n130_aliasMSR: 2202_9600h
Init: _ccd4_lthree0_core1_thread1_inst3_n131_aliasMSR: 2202_9700h
Init: _ccd4_lthree0_core2_thread0_inst3_n132_aliasMSR: 2204_9600h
Init: _ccd4_lthree0_core2_thread1_inst3_n133_aliasMSR: 2204_9700h
Init: _ccd4_lthree0_core3_thread0_inst3_n134_aliasMSR: 2206_9600h
Init: _ccd4_lthree0_core3_thread1_inst3_n135_aliasMSR: 2206_9700h
Init: _ccd4_lthree0_core4_thread0_inst3_n136_aliasMSR: 2208_9600h
Init: _ccd4_lthree0_core4_thread1_inst3_n137_aliasMSR: 2208_9700h
Init: _ccd4_lthree0_core5_thread0_inst3_n138_aliasMSR: 220A_9600h
Init: _ccd4_lthree0_core5_thread1_inst3_n139_aliasMSR: 220A_9700h
Init: _ccd4_lthree0_core6_thread0_inst3_n140_aliasMSR: 220C_9600h
Init: _ccd4_lthree0_core6_thread1_inst3_n141_aliasMSR: 220C_9700h
Init: _ccd4_lthree0_core7_thread0_inst3_n142_aliasMSR: 220E_9600h
Init: _ccd4_lthree0_core7_thread1_inst3_n143_aliasMSR: 220E_9700h
Init: _ccd4_lthree0_core8_thread0_inst3_n144_aliasMSR: 2210_9600h
Init: _ccd4_lthree0_core8_thread1_inst3_n145_aliasMSR: 2210_9700h
Init: _ccd4_lthree0_core9_thread0_inst3_n146_aliasMSR: 2212_9600h
Init: _ccd4_lthree0_core9_thread1_inst3_n147_aliasMSR: 2212_9700h
Init: _ccd4_lthree0_core10_thread0_inst3_n148_aliasMSR: 2214_9600h
Init: _ccd4_lthree0_core10_thread1_inst3_n149_aliasMSR: 2214_9700h
Init: _ccd4_lthree0_core11_thread0_inst3_n150_aliasMSR: 2216_9600h
Init: _ccd4_lthree0_core11_thread1_inst3_n151_aliasMSR: 2216_9700h
Init: _ccd4_lthree0_core12_thread0_inst3_n152_aliasMSR: 2218_9600h
Init: _ccd4_lthree0_core12_thread1_inst3_n153_aliasMSR: 2218_9700h
Init: _ccd4_lthree0_core13_thread0_inst3_n154_aliasMSR: 221A_9600h
Init: _ccd4_lthree0_core13_thread1_inst3_n155_aliasMSR: 221A_9700h
Init: _ccd4_lthree0_core14_thread0_inst3_n156_aliasMSR: 221C_9600h
Init: _ccd4_lthree0_core14_thread1_inst3_n157_aliasMSR: 221C_9700h
Init: _ccd4_lthree0_core15_thread0_inst3_n158_aliasMSR: 221E_9600h
Init: _ccd4_lthree0_core15_thread1_inst3_n159_aliasMSR: 221E_9700h
Init: _ccd5_lthree0_core0_thread0_inst3_n160_aliasMSR: 2280_9600h
Init: _ccd5_lthree0_core0_thread1_inst3_n161_aliasMSR: 2280_9700h
Init: _ccd5_lthree0_core1_thread0_inst3_n162_aliasMSR: 2282_9600h
Init: _ccd5_lthree0_core1_thread1_inst3_n163_aliasMSR: 2282_9700h
Init: _ccd5_lthree0_core2_thread0_inst3_n164_aliasMSR: 2284_9600h
Init: _ccd5_lthree0_core2_thread1_inst3_n165_aliasMSR: 2284_9700h
Init: _ccd5_lthree0_core3_thread0_inst3_n166_aliasMSR: 2286_9600h
Init: _ccd5_lthree0_core3_thread1_inst3_n167_aliasMSR: 2286_9700h
Init: _ccd5_lthree0_core4_thread0_inst3_n168_aliasMSR: 2288_9600h
Init: _ccd5_lthree0_core4_thread1_inst3_n169_aliasMSR: 2288_9700h
Init: _ccd5_lthree0_core5_thread0_inst3_n170_aliasMSR: 228A_9600h
Init: _ccd5_lthree0_core5_thread1_inst3_n171_aliasMSR: 228A_9700h
Init: _ccd5_lthree0_core6_thread0_inst3_n172_aliasMSR: 228C_9600h
Init: _ccd5_lthree0_core6_thread1_inst3_n173_aliasMSR: 228C_9700h
Init: _ccd5_lthree0_core7_thread0_inst3_n174_aliasMSR: 228E_9600h
Init: _ccd5_lthree0_core7_thread1_inst3_n175_aliasMSR: 228E_9700h
Init: _ccd5_lthree0_core8_thread0_inst3_n176_aliasMSR: 2290_9600h
Init: _ccd5_lthree0_core8_thread1_inst3_n177_aliasMSR: 2290_9700h
Init: _ccd5_lthree0_core9_thread0_inst3_n178_aliasMSR: 2292_9600h

Init: _ccd5_lthree0_core9_thread1_inst3_n179_aliasMSR: 2292_9700h
Init: _ccd5_lthree0_core10_thread0_inst3_n180_aliasMSR: 2294_9600h
Init: _ccd5_lthree0_core10_thread1_inst3_n181_aliasMSR: 2294_9700h
Init: _ccd5_lthree0_core11_thread0_inst3_n182_aliasMSR: 2296_9600h
Init: _ccd5_lthree0_core11_thread1_inst3_n183_aliasMSR: 2296_9700h
Init: _ccd5_lthree0_core12_thread0_inst3_n184_aliasMSR: 2298_9600h
Init: _ccd5_lthree0_core12_thread1_inst3_n185_aliasMSR: 2298_9700h
Init: _ccd5_lthree0_core13_thread0_inst3_n186_aliasMSR: 229A_9600h
Init: _ccd5_lthree0_core13_thread1_inst3_n187_aliasMSR: 229A_9700h
Init: _ccd5_lthree0_core14_thread0_inst3_n188_aliasMSR: 229C_9600h
Init: _ccd5_lthree0_core14_thread1_inst3_n189_aliasMSR: 229C_9700h
Init: _ccd5_lthree0_core15_thread0_inst3_n190_aliasMSR: 229E_9600h
Init: _ccd5_lthree0_core15_thread1_inst3_n191_aliasMSR: 229E_9700h
Init: _ccd6_lthree0_core0_thread0_inst3_n192_aliasMSR: 2300_9600h
Init: _ccd6_lthree0_core0_thread1_inst3_n193_aliasMSR: 2300_9700h
Init: _ccd6_lthree0_core1_thread0_inst3_n194_aliasMSR: 2302_9600h
Init: _ccd6_lthree0_core1_thread1_inst3_n195_aliasMSR: 2302_9700h
Init: _ccd6_lthree0_core2_thread0_inst3_n196_aliasMSR: 2304_9600h
Init: _ccd6_lthree0_core2_thread1_inst3_n197_aliasMSR: 2304_9700h
Init: _ccd6_lthree0_core3_thread0_inst3_n198_aliasMSR: 2306_9600h
Init: _ccd6_lthree0_core3_thread1_inst3_n199_aliasMSR: 2306_9700h
Init: _ccd6_lthree0_core4_thread0_inst3_n200_aliasMSR: 2308_9600h
Init: _ccd6_lthree0_core4_thread1_inst3_n201_aliasMSR: 2308_9700h
Init: _ccd6_lthree0_core5_thread0_inst3_n202_aliasMSR: 230A_9600h
Init: _ccd6_lthree0_core5_thread1_inst3_n203_aliasMSR: 230A_9700h
Init: _ccd6_lthree0_core6_thread0_inst3_n204_aliasMSR: 230C_9600h
Init: _ccd6_lthree0_core6_thread1_inst3_n205_aliasMSR: 230C_9700h
Init: _ccd6_lthree0_core7_thread0_inst3_n206_aliasMSR: 230E_9600h
Init: _ccd6_lthree0_core7_thread1_inst3_n207_aliasMSR: 230E_9700h
Init: _ccd6_lthree0_core8_thread0_inst3_n208_aliasMSR: 2310_9600h
Init: _ccd6_lthree0_core8_thread1_inst3_n209_aliasMSR: 2310_9700h
Init: _ccd6_lthree0_core9_thread0_inst3_n210_aliasMSR: 2312_9600h
Init: _ccd6_lthree0_core9_thread1_inst3_n211_aliasMSR: 2312_9700h
Init: _ccd6_lthree0_core10_thread0_inst3_n212_aliasMSR: 2314_9600h
Init: _ccd6_lthree0_core10_thread1_inst3_n213_aliasMSR: 2314_9700h
Init: _ccd6_lthree0_core11_thread0_inst3_n214_aliasMSR: 2316_9600h
Init: _ccd6_lthree0_core11_thread1_inst3_n215_aliasMSR: 2316_9700h
Init: _ccd6_lthree0_core12_thread0_inst3_n216_aliasMSR: 2318_9600h
Init: _ccd6_lthree0_core12_thread1_inst3_n217_aliasMSR: 2318_9700h
Init: _ccd6_lthree0_core13_thread0_inst3_n218_aliasMSR: 231A_9600h
Init: _ccd6_lthree0_core13_thread1_inst3_n219_aliasMSR: 231A_9700h
Init: _ccd6_lthree0_core14_thread0_inst3_n220_aliasMSR: 231C_9600h
Init: _ccd6_lthree0_core14_thread1_inst3_n221_aliasMSR: 231C_9700h
Init: _ccd6_lthree0_core15_thread0_inst3_n222_aliasMSR: 231E_9600h
Init: _ccd6_lthree0_core15_thread1_inst3_n223_aliasMSR: 231E_9700h
Init: _ccd7_lthree0_core0_thread0_inst3_n224_aliasMSR: 2380_9600h
Init: _ccd7_lthree0_core0_thread1_inst3_n225_aliasMSR: 2380_9700h
Init: _ccd7_lthree0_core1_thread0_inst3_n226_aliasMSR: 2382_9600h
Init: _ccd7_lthree0_core1_thread1_inst3_n227_aliasMSR: 2382_9700h
Init: _ccd7_lthree0_core2_thread0_inst3_n228_aliasMSR: 2384_9600h
Init: _ccd7_lthree0_core2_thread1_inst3_n229_aliasMSR: 2384_9700h
Init: _ccd7_lthree0_core3_thread0_inst3_n230_aliasMSR: 2386_9600h
Init: _ccd7_lthree0_core3_thread1_inst3_n231_aliasMSR: 2386_9700h
Init: _ccd7_lthree0_core4_thread0_inst3_n232_aliasMSR: 2388_9600h
Init: _ccd7_lthree0_core4_thread1_inst3_n233_aliasMSR: 2388_9700h
Init: _ccd7_lthree0_core5_thread0_inst3_n234_aliasMSR: 238A_9600h
Init: _ccd7_lthree0_core5_thread1_inst3_n235_aliasMSR: 238A_9700h
Init: _ccd7_lthree0_core6_thread0_inst3_n236_aliasMSR: 238C_9600h
Init: _ccd7_lthree0_core6_thread1_inst3_n237_aliasMSR: 238C_9700h
Init: _ccd7_lthree0_core7_thread0_inst3_n238_aliasMSR: 238E_9600h
Init: _ccd7_lthree0_core7_thread1_inst3_n239_aliasMSR: 238E_9700h
Init: _ccd7_lthree0_core8_thread0_inst3_n240_aliasMSR: 2390_9600h
Init: _ccd7_lthree0_core8_thread1_inst3_n241_aliasMSR: 2390_9700h
Init: _ccd7_lthree0_core9_thread0_inst3_n242_aliasMSR: 2392_9600h
Init: _ccd7_lthree0_core9_thread1_inst3_n243_aliasMSR: 2392_9700h

Init: _ccd7_lthree0_core10_thread0_inst3_n244_aliasMSR: 2394_9600h
Init: _ccd7_lthree0_core10_thread1_inst3_n245_aliasMSR: 2394_9700h
Init: _ccd7_lthree0_core11_thread0_inst3_n246_aliasMSR: 2396_9600h
Init: _ccd7_lthree0_core11_thread1_inst3_n247_aliasMSR: 2396_9700h
Init: _ccd7_lthree0_core12_thread0_inst3_n248_aliasMSR: 2398_9600h
Init: _ccd7_lthree0_core12_thread1_inst3_n249_aliasMSR: 2398_9700h
Init: _ccd7_lthree0_core13_thread0_inst3_n250_aliasMSR: 239A_9600h
Init: _ccd7_lthree0_core13_thread1_inst3_n251_aliasMSR: 239A_9700h
Init: _ccd7_lthree0_core14_thread0_inst3_n252_aliasMSR: 239C_9600h
Init: _ccd7_lthree0_core14_thread1_inst3_n253_aliasMSR: 239C_9700h
Init: _ccd7_lthree0_core15_thread0_inst3_n254_aliasMSR: 239E_9600h
Init: _ccd7_lthree0_core15_thread1_inst3_n255_aliasMSR: 239E_9700h
Init: _ccd8_lthree0_core0_thread0_inst3_n256_aliasMSR: 2400_9600h
Init: _ccd8_lthree0_core0_thread1_inst3_n257_aliasMSR: 2400_9700h
Init: _ccd8_lthree0_core1_thread0_inst3_n258_aliasMSR: 2402_9600h
Init: _ccd8_lthree0_core1_thread1_inst3_n259_aliasMSR: 2402_9700h
Init: _ccd8_lthree0_core2_thread0_inst3_n260_aliasMSR: 2404_9600h
Init: _ccd8_lthree0_core2_thread1_inst3_n261_aliasMSR: 2404_9700h
Init: _ccd8_lthree0_core3_thread0_inst3_n262_aliasMSR: 2406_9600h
Init: _ccd8_lthree0_core3_thread1_inst3_n263_aliasMSR: 2406_9700h
Init: _ccd8_lthree0_core4_thread0_inst3_n264_aliasMSR: 2408_9600h
Init: _ccd8_lthree0_core4_thread1_inst3_n265_aliasMSR: 2408_9700h
Init: _ccd8_lthree0_core5_thread0_inst3_n266_aliasMSR: 240A_9600h
Init: _ccd8_lthree0_core5_thread1_inst3_n267_aliasMSR: 240A_9700h
Init: _ccd8_lthree0_core6_thread0_inst3_n268_aliasMSR: 240C_9600h
Init: _ccd8_lthree0_core6_thread1_inst3_n269_aliasMSR: 240C_9700h
Init: _ccd8_lthree0_core7_thread0_inst3_n270_aliasMSR: 240E_9600h
Init: _ccd8_lthree0_core7_thread1_inst3_n271_aliasMSR: 240E_9700h
Init: _ccd8_lthree0_core8_thread0_inst3_n272_aliasMSR: 2410_9600h
Init: _ccd8_lthree0_core8_thread1_inst3_n273_aliasMSR: 2410_9700h
Init: _ccd8_lthree0_core9_thread0_inst3_n274_aliasMSR: 2412_9600h
Init: _ccd8_lthree0_core9_thread1_inst3_n275_aliasMSR: 2412_9700h
Init: _ccd8_lthree0_core10_thread0_inst3_n276_aliasMSR: 2414_9600h
Init: _ccd8_lthree0_core10_thread1_inst3_n277_aliasMSR: 2414_9700h
Init: _ccd8_lthree0_core11_thread0_inst3_n278_aliasMSR: 2416_9600h
Init: _ccd8_lthree0_core11_thread1_inst3_n279_aliasMSR: 2416_9700h
Init: _ccd8_lthree0_core12_thread0_inst3_n280_aliasMSR: 2418_9600h
Init: _ccd8_lthree0_core12_thread1_inst3_n281_aliasMSR: 2418_9700h
Init: _ccd8_lthree0_core13_thread0_inst3_n282_aliasMSR: 241A_9600h
Init: _ccd8_lthree0_core13_thread1_inst3_n283_aliasMSR: 241A_9700h
Init: _ccd8_lthree0_core14_thread0_inst3_n284_aliasMSR: 241C_9600h
Init: _ccd8_lthree0_core14_thread1_inst3_n285_aliasMSR: 241C_9700h
Init: _ccd8_lthree0_core15_thread0_inst3_n286_aliasMSR: 241E_9600h
Init: _ccd8_lthree0_core15_thread1_inst3_n287_aliasMSR: 241E_9700h
Init: _ccd9_lthree0_core0_thread0_inst3_n288_aliasMSR: 2480_9600h
Init: _ccd9_lthree0_core0_thread1_inst3_n289_aliasMSR: 2480_9700h
Init: _ccd9_lthree0_core1_thread0_inst3_n290_aliasMSR: 2482_9600h
Init: _ccd9_lthree0_core1_thread1_inst3_n291_aliasMSR: 2482_9700h
Init: _ccd9_lthree0_core2_thread0_inst3_n292_aliasMSR: 2484_9600h
Init: _ccd9_lthree0_core2_thread1_inst3_n293_aliasMSR: 2484_9700h
Init: _ccd9_lthree0_core3_thread0_inst3_n294_aliasMSR: 2486_9600h
Init: _ccd9_lthree0_core3_thread1_inst3_n295_aliasMSR: 2486_9700h
Init: _ccd9_lthree0_core4_thread0_inst3_n296_aliasMSR: 2488_9600h
Init: _ccd9_lthree0_core4_thread1_inst3_n297_aliasMSR: 2488_9700h
Init: _ccd9_lthree0_core5_thread0_inst3_n298_aliasMSR: 248A_9600h
Init: _ccd9_lthree0_core5_thread1_inst3_n299_aliasMSR: 248A_9700h
Init: _ccd9_lthree0_core6_thread0_inst3_n300_aliasMSR: 248C_9600h
Init: _ccd9_lthree0_core6_thread1_inst3_n301_aliasMSR: 248C_9700h
Init: _ccd9_lthree0_core7_thread0_inst3_n302_aliasMSR: 248E_9600h
Init: _ccd9_lthree0_core7_thread1_inst3_n303_aliasMSR: 248E_9700h
Init: _ccd9_lthree0_core8_thread0_inst3_n304_aliasMSR: 2490_9600h
Init: _ccd9_lthree0_core8_thread1_inst3_n305_aliasMSR: 2490_9700h
Init: _ccd9_lthree0_core9_thread0_inst3_n306_aliasMSR: 2492_9600h
Init: _ccd9_lthree0_core9_thread1_inst3_n307_aliasMSR: 2492_9700h
Init: _ccd9_lthree0_core10_thread0_inst3_n308_aliasMSR: 2494_9600h

Init: _ccd9_lthree0_core10_thread1_inst3_n309_aliasMSR: 2494_9700h
Init: _ccd9_lthree0_core11_thread0_inst3_n310_aliasMSR: 2496_9600h
Init: _ccd9_lthree0_core11_thread1_inst3_n311_aliasMSR: 2496_9700h
Init: _ccd9_lthree0_core12_thread0_inst3_n312_aliasMSR: 2498_9600h
Init: _ccd9_lthree0_core12_thread1_inst3_n313_aliasMSR: 2498_9700h
Init: _ccd9_lthree0_core13_thread0_inst3_n314_aliasMSR: 249A_9600h
Init: _ccd9_lthree0_core13_thread1_inst3_n315_aliasMSR: 249A_9700h
Init: _ccd9_lthree0_core14_thread0_inst3_n316_aliasMSR: 249C_9600h
Init: _ccd9_lthree0_core14_thread1_inst3_n317_aliasMSR: 249C_9700h
Init: _ccd9_lthree0_core15_thread0_inst3_n318_aliasMSR: 249E_9600h
Init: _ccd9_lthree0_core15_thread1_inst3_n319_aliasMSR: 249E_9700h
Init: _ccd10_lthree0_core0_thread0_inst3_n320_aliasMSR: 2500_9600h
Init: _ccd10_lthree0_core0_thread1_inst3_n321_aliasMSR: 2500_9700h
Init: _ccd10_lthree0_core1_thread0_inst3_n322_aliasMSR: 2502_9600h
Init: _ccd10_lthree0_core1_thread1_inst3_n323_aliasMSR: 2502_9700h
Init: _ccd10_lthree0_core2_thread0_inst3_n324_aliasMSR: 2504_9600h
Init: _ccd10_lthree0_core2_thread1_inst3_n325_aliasMSR: 2504_9700h
Init: _ccd10_lthree0_core3_thread0_inst3_n326_aliasMSR: 2506_9600h
Init: _ccd10_lthree0_core3_thread1_inst3_n327_aliasMSR: 2506_9700h
Init: _ccd10_lthree0_core4_thread0_inst3_n328_aliasMSR: 2508_9600h
Init: _ccd10_lthree0_core4_thread1_inst3_n329_aliasMSR: 2508_9700h
Init: _ccd10_lthree0_core5_thread0_inst3_n330_aliasMSR: 250A_9600h
Init: _ccd10_lthree0_core5_thread1_inst3_n331_aliasMSR: 250A_9700h
Init: _ccd10_lthree0_core6_thread0_inst3_n332_aliasMSR: 250C_9600h
Init: _ccd10_lthree0_core6_thread1_inst3_n333_aliasMSR: 250C_9700h
Init: _ccd10_lthree0_core7_thread0_inst3_n334_aliasMSR: 250E_9600h
Init: _ccd10_lthree0_core7_thread1_inst3_n335_aliasMSR: 250E_9700h
Init: _ccd10_lthree0_core8_thread0_inst3_n336_aliasMSR: 2510_9600h
Init: _ccd10_lthree0_core8_thread1_inst3_n337_aliasMSR: 2510_9700h
Init: _ccd10_lthree0_core9_thread0_inst3_n338_aliasMSR: 2512_9600h
Init: _ccd10_lthree0_core9_thread1_inst3_n339_aliasMSR: 2512_9700h
Init: _ccd10_lthree0_core10_thread0_inst3_n340_aliasMSR: 2514_9600h
Init: _ccd10_lthree0_core10_thread1_inst3_n341_aliasMSR: 2514_9700h
Init: _ccd10_lthree0_core11_thread0_inst3_n342_aliasMSR: 2516_9600h
Init: _ccd10_lthree0_core11_thread1_inst3_n343_aliasMSR: 2516_9700h
Init: _ccd10_lthree0_core12_thread0_inst3_n344_aliasMSR: 2518_9600h
Init: _ccd10_lthree0_core12_thread1_inst3_n345_aliasMSR: 2518_9700h
Init: _ccd10_lthree0_core13_thread0_inst3_n346_aliasMSR: 251A_9600h
Init: _ccd10_lthree0_core13_thread1_inst3_n347_aliasMSR: 251A_9700h
Init: _ccd10_lthree0_core14_thread0_inst3_n348_aliasMSR: 251C_9600h
Init: _ccd10_lthree0_core14_thread1_inst3_n349_aliasMSR: 251C_9700h
Init: _ccd10_lthree0_core15_thread0_inst3_n350_aliasMSR: 251E_9600h
Init: _ccd10_lthree0_core15_thread1_inst3_n351_aliasMSR: 251E_9700h
Init: _ccd11_lthree0_core0_thread0_inst3_n352_aliasMSR: 2580_9600h
Init: _ccd11_lthree0_core0_thread1_inst3_n353_aliasMSR: 2580_9700h
Init: _ccd11_lthree0_core1_thread0_inst3_n354_aliasMSR: 2582_9600h
Init: _ccd11_lthree0_core1_thread1_inst3_n355_aliasMSR: 2582_9700h
Init: _ccd11_lthree0_core2_thread0_inst3_n356_aliasMSR: 2584_9600h
Init: _ccd11_lthree0_core2_thread1_inst3_n357_aliasMSR: 2584_9700h
Init: _ccd11_lthree0_core3_thread0_inst3_n358_aliasMSR: 2586_9600h
Init: _ccd11_lthree0_core3_thread1_inst3_n359_aliasMSR: 2586_9700h
Init: _ccd11_lthree0_core4_thread0_inst3_n360_aliasMSR: 2588_9600h
Init: _ccd11_lthree0_core4_thread1_inst3_n361_aliasMSR: 2588_9700h
Init: _ccd11_lthree0_core5_thread0_inst3_n362_aliasMSR: 258A_9600h
Init: _ccd11_lthree0_core5_thread1_inst3_n363_aliasMSR: 258A_9700h
Init: _ccd11_lthree0_core6_thread0_inst3_n364_aliasMSR: 258C_9600h
Init: _ccd11_lthree0_core6_thread1_inst3_n365_aliasMSR: 258C_9700h
Init: _ccd11_lthree0_core7_thread0_inst3_n366_aliasMSR: 258E_9600h
Init: _ccd11_lthree0_core7_thread1_inst3_n367_aliasMSR: 258E_9700h
Init: _ccd11_lthree0_core8_thread0_inst3_n368_aliasMSR: 2590_9600h
Init: _ccd11_lthree0_core8_thread1_inst3_n369_aliasMSR: 2590_9700h
Init: _ccd11_lthree0_core9_thread0_inst3_n370_aliasMSR: 2592_9600h
Init: _ccd11_lthree0_core9_thread1_inst3_n371_aliasMSR: 2592_9700h
Init: _ccd11_lthree0_core10_thread0_inst3_n372_aliasMSR: 2594_9600h
Init: _ccd11_lthree0_core10_thread1_inst3_n373_aliasMSR: 2594_9700h

Init: _ccd11_lthree0_core11_thread0_inst3_n374_aliasMSR: 2596_9600h
Init: _ccd11_lthree0_core11_thread1_inst3_n375_aliasMSR: 2596_9700h
Init: _ccd11_lthree0_core12_thread0_inst3_n376_aliasMSR: 2598_9600h
Init: _ccd11_lthree0_core12_thread1_inst3_n377_aliasMSR: 2598_9700h
Init: _ccd11_lthree0_core13_thread0_inst3_n378_aliasMSR: 259A_9600h
Init: _ccd11_lthree0_core13_thread1_inst3_n379_aliasMSR: 259A_9700h
Init: _ccd11_lthree0_core14_thread0_inst3_n380_aliasMSR: 259C_9600h
Init: _ccd11_lthree0_core14_thread1_inst3_n381_aliasMSR: 259C_9700h
Init: _ccd11_lthree0_core15_thread0_inst3_n382_aliasMSR: 259E_9600h
Init: _ccd11_lthree0_core15_thread1_inst3_n383_aliasMSR: 259E_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](#)

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst3_n[383:0]_aliasMSR; MSRC000_2036

Bits	Description
63:32	Syndrom . 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	ErrorPriority . 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	Length . Read-write, Volatile . Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::DE::MCA_SYND_DE[Syndrome]. Length values greater than 32 (decimal) are interpreted as equal to 32 (decimal). 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	ErrorInformation . 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_DE].

Table 63: 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
lbq	[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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst3_n[383:0]_aliasMSR; MSRC001_0403

Bits	Description
63:10	Reserved.
9	HWA. Read-write. Reset: 0. Hardware Assertion Error.
8	OCQ. Read-write. Reset: 0. Micro-op fetch queue parity error.
7	UcSeq. Read-write. Reset: 0. Patch RAM sequencer parity error.
6	UcDat. Read-write. Reset: 0. Patch RAM data parity error.
5	Faq. Read-write. Reset: 0. Fetch address FIFO parity error.
4	Idq. Read-write. Reset: 0. Instruction dispatch queue parity error.
3	UopQ. Read-write. Reset: 0. Micro-op queue parity error.
2	Ibq. Read-write. Reset: 0. Instruction buffer parity error.
1	OcDat. Read-write. Reset: 0. Micro-op cache data parity error.
0	OcTag. 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](#)

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst3_n[383:0]_aliasMSR; MSRC000_203E

Bits	Description
63:0	Syndrome. 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](#)

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst3_n[383:0]_aliasMSR; MSRC000_203F

Bits	Description
63:0	Syndrome. 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst5_n[383:0]_aliasMSRLEGACY; MSR0000_0414

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst5_n[383:0]_aliasMSR; MSRC000_2050

Bits	Description
63:14	Reserved.
13	RETMAP . Read-write. Reset: 0. Retire Map parity error.
12	SPECMAP . Read-write. Reset: 0. PRN/FRN freelist parity error.
11	HWA . Read-write. Reset: 0. Hardware Assertion error.
10	BBQ . Read-write. Reset: 0. Branch buffer queue parity error.
9	SQ . Read-write. Reset: 0. EXTID parity error.
8	STATQ . Read-write. Reset: 0. Retire status queue parity error.
7	RETDISP . Read-write. Reset: 0. Retire dispatch queue parity error.
6	CHKPTQ . Read-write. Reset: 0. CHKPTQ. Checkpoint queue parity error.
5	PLDAL . Read-write. Reset: 0. EX payload parity error.
4	PLDAG . Read-write. Reset: 0. Address generator payload parity error.
3	IDRF . Read-write. Reset: 0. Immediate displacement register file parity error.
2	FRF . Read-write. Reset: 0. Flag register file parity error.
1	PRF . Read-write. Reset: 0. Physical register file parity error.
0	WDT . 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst5_n[383:0]_aliasMSRLEGACY; MSR0000_0415

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst5_n[383:0]_aliasMSR; MSRC000_2051

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. 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	MiscV. 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	AddrV. 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	PCC. 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	ErrCoreIdVal. 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	TCC. 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	RESERV54. 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	SyndV. 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	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. 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	RESERV22. 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	ErrorCodeExt. 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	ErrorCode. 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 64: 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst5_n[383:0]_aliasMSRLEGACY; MSR0000_0416

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst5_n[383:0]_aliasMSR; MSRC000_2052

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::EX::MCA_STATUS_EX. For physical addresses, the most significant bit is given by Core::X86::CpuId::LongModeInfo[PhysAddrSize].

Table 65: 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst5_n[383:0]_aliasMSRLEGACY; MSR0000_0417

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst5_n[383:0]_aliasMSR; MSRC000_2053

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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::EX::MCA_MISC0_EX[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	LvtOffset. 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::EX::MCA_MISC0_EX[Locked]) ? Read-write : Read-only.
51	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msrr::HWCR[McStatusWrEn] !MCA::EX::MCA_MISC0_EX[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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::EX::MCA_MISC0_EX[Locked]) ? Read-write : Read-only.
48	Ovrflw. 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::EX::MCA_MISC0_EX[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	ErrCnt. 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::EX::MCA_MISC0_EX[Locked]) ? Read-write : Read-only.
31:24	BlkPtr. 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst5_n[383:0]_aliasMSR; MSRC000_2054

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	McaXEnable. 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	IntPresent. 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	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::EX::MCA_CONFIG_EX[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. 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	DeferredErrorLoggingSupported. 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	McaX. 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst5_n[383:0]_aliasMSR; MSRC000_2055

Bits	Description
63:48	McaType . Read-only. Reset: 0005h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi . 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	HardwareID . Read-only. Reset: 0B0h. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _ccd0_lthree0_core0_thread0_inst5_n0_aliasMSR: 2000_9A00h
	Init: _ccd0_lthree0_core0_thread1_inst5_n1_aliasMSR: 2000_9B00h
	Init: _ccd0_lthree0_core1_thread0_inst5_n2_aliasMSR: 2002_9A00h
	Init: _ccd0_lthree0_core1_thread1_inst5_n3_aliasMSR: 2002_9B00h
	Init: _ccd0_lthree0_core2_thread0_inst5_n4_aliasMSR: 2004_9A00h
	Init: _ccd0_lthree0_core2_thread1_inst5_n5_aliasMSR: 2004_9B00h
	Init: _ccd0_lthree0_core3_thread0_inst5_n6_aliasMSR: 2006_9A00h
	Init: _ccd0_lthree0_core3_thread1_inst5_n7_aliasMSR: 2006_9B00h
	Init: _ccd0_lthree0_core4_thread0_inst5_n8_aliasMSR: 2008_9A00h
	Init: _ccd0_lthree0_core4_thread1_inst5_n9_aliasMSR: 2008_9B00h
	Init: _ccd0_lthree0_core5_thread0_inst5_n10_aliasMSR: 200A_9A00h
	Init: _ccd0_lthree0_core5_thread1_inst5_n11_aliasMSR: 200A_9B00h
	Init: _ccd0_lthree0_core6_thread0_inst5_n12_aliasMSR: 200C_9A00h
	Init: _ccd0_lthree0_core6_thread1_inst5_n13_aliasMSR: 200C_9B00h
	Init: _ccd0_lthree0_core7_thread0_inst5_n14_aliasMSR: 200E_9A00h
	Init: _ccd0_lthree0_core7_thread1_inst5_n15_aliasMSR: 200E_9B00h
	Init: _ccd0_lthree0_core8_thread0_inst5_n16_aliasMSR: 2010_9A00h
	Init: _ccd0_lthree0_core8_thread1_inst5_n17_aliasMSR: 2010_9B00h
	Init: _ccd0_lthree0_core9_thread0_inst5_n18_aliasMSR: 2012_9A00h
	Init: _ccd0_lthree0_core9_thread1_inst5_n19_aliasMSR: 2012_9B00h
	Init: _ccd0_lthree0_core10_thread0_inst5_n20_aliasMSR: 2014_9A00h
	Init: _ccd0_lthree0_core10_thread1_inst5_n21_aliasMSR: 2014_9B00h
	Init: _ccd0_lthree0_core11_thread0_inst5_n22_aliasMSR: 2016_9A00h
	Init: _ccd0_lthree0_core11_thread1_inst5_n23_aliasMSR: 2016_9B00h
	Init: _ccd0_lthree0_core12_thread0_inst5_n24_aliasMSR: 2018_9A00h
	Init: _ccd0_lthree0_core12_thread1_inst5_n25_aliasMSR: 2018_9B00h
	Init: _ccd0_lthree0_core13_thread0_inst5_n26_aliasMSR: 201A_9A00h
	Init: _ccd0_lthree0_core13_thread1_inst5_n27_aliasMSR: 201A_9B00h
	Init: _ccd0_lthree0_core14_thread0_inst5_n28_aliasMSR: 201C_9A00h
	Init: _ccd0_lthree0_core14_thread1_inst5_n29_aliasMSR: 201C_9B00h
	Init: _ccd0_lthree0_core15_thread0_inst5_n30_aliasMSR: 201E_9A00h
	Init: _ccd0_lthree0_core15_thread1_inst5_n31_aliasMSR: 201E_9B00h
	Init: _ccd1_lthree0_core0_thread0_inst5_n32_aliasMSR: 2080_9A00h
	Init: _ccd1_lthree0_core0_thread1_inst5_n33_aliasMSR: 2080_9B00h
	Init: _ccd1_lthree0_core1_thread0_inst5_n34_aliasMSR: 2082_9A00h
	Init: _ccd1_lthree0_core1_thread1_inst5_n35_aliasMSR: 2082_9B00h
	Init: _ccd1_lthree0_core2_thread0_inst5_n36_aliasMSR: 2084_9A00h
	Init: _ccd1_lthree0_core2_thread1_inst5_n37_aliasMSR: 2084_9B00h
	Init: _ccd1_lthree0_core3_thread0_inst5_n38_aliasMSR: 2086_9A00h
	Init: _ccd1_lthree0_core3_thread1_inst5_n39_aliasMSR: 2086_9B00h
	Init: _ccd1_lthree0_core4_thread0_inst5_n40_aliasMSR: 2088_9A00h
	Init: _ccd1_lthree0_core4_thread1_inst5_n41_aliasMSR: 2088_9B00h
	Init: _ccd1_lthree0_core5_thread0_inst5_n42_aliasMSR: 208A_9A00h
	Init: _ccd1_lthree0_core5_thread1_inst5_n43_aliasMSR: 208A_9B00h
	Init: _ccd1_lthree0_core6_thread0_inst5_n44_aliasMSR: 208C_9A00h
	Init: _ccd1_lthree0_core6_thread1_inst5_n45_aliasMSR: 208C_9B00h
	Init: _ccd1_lthree0_core7_thread0_inst5_n46_aliasMSR: 208E_9A00h
	Init: _ccd1_lthree0_core7_thread1_inst5_n47_aliasMSR: 208E_9B00h
	Init: _ccd1_lthree0_core8_thread0_inst5_n48_aliasMSR: 2090_9A00h

Init: _ccd1_lthree0_core8_thread1_inst5_n49_aliasMSR: 2090_9B00h
Init: _ccd1_lthree0_core9_thread0_inst5_n50_aliasMSR: 2092_9A00h
Init: _ccd1_lthree0_core9_thread1_inst5_n51_aliasMSR: 2092_9B00h
Init: _ccd1_lthree0_core10_thread0_inst5_n52_aliasMSR: 2094_9A00h
Init: _ccd1_lthree0_core10_thread1_inst5_n53_aliasMSR: 2094_9B00h
Init: _ccd1_lthree0_core11_thread0_inst5_n54_aliasMSR: 2096_9A00h
Init: _ccd1_lthree0_core11_thread1_inst5_n55_aliasMSR: 2096_9B00h
Init: _ccd1_lthree0_core12_thread0_inst5_n56_aliasMSR: 2098_9A00h
Init: _ccd1_lthree0_core12_thread1_inst5_n57_aliasMSR: 2098_9B00h
Init: _ccd1_lthree0_core13_thread0_inst5_n58_aliasMSR: 209A_9A00h
Init: _ccd1_lthree0_core13_thread1_inst5_n59_aliasMSR: 209A_9B00h
Init: _ccd1_lthree0_core14_thread0_inst5_n60_aliasMSR: 209C_9A00h
Init: _ccd1_lthree0_core14_thread1_inst5_n61_aliasMSR: 209C_9B00h
Init: _ccd1_lthree0_core15_thread0_inst5_n62_aliasMSR: 209E_9A00h
Init: _ccd1_lthree0_core15_thread1_inst5_n63_aliasMSR: 209E_9B00h
Init: _ccd2_lthree0_core0_thread0_inst5_n64_aliasMSR: 2100_9A00h
Init: _ccd2_lthree0_core0_thread1_inst5_n65_aliasMSR: 2100_9B00h
Init: _ccd2_lthree0_core1_thread0_inst5_n66_aliasMSR: 2102_9A00h
Init: _ccd2_lthree0_core1_thread1_inst5_n67_aliasMSR: 2102_9B00h
Init: _ccd2_lthree0_core2_thread0_inst5_n68_aliasMSR: 2104_9A00h
Init: _ccd2_lthree0_core2_thread1_inst5_n69_aliasMSR: 2104_9B00h
Init: _ccd2_lthree0_core3_thread0_inst5_n70_aliasMSR: 2106_9A00h
Init: _ccd2_lthree0_core3_thread1_inst5_n71_aliasMSR: 2106_9B00h
Init: _ccd2_lthree0_core4_thread0_inst5_n72_aliasMSR: 2108_9A00h
Init: _ccd2_lthree0_core4_thread1_inst5_n73_aliasMSR: 2108_9B00h
Init: _ccd2_lthree0_core5_thread0_inst5_n74_aliasMSR: 210A_9A00h
Init: _ccd2_lthree0_core5_thread1_inst5_n75_aliasMSR: 210A_9B00h
Init: _ccd2_lthree0_core6_thread0_inst5_n76_aliasMSR: 210C_9A00h
Init: _ccd2_lthree0_core6_thread1_inst5_n77_aliasMSR: 210C_9B00h
Init: _ccd2_lthree0_core7_thread0_inst5_n78_aliasMSR: 210E_9A00h
Init: _ccd2_lthree0_core7_thread1_inst5_n79_aliasMSR: 210E_9B00h
Init: _ccd2_lthree0_core8_thread0_inst5_n80_aliasMSR: 2110_9A00h
Init: _ccd2_lthree0_core8_thread1_inst5_n81_aliasMSR: 2110_9B00h
Init: _ccd2_lthree0_core9_thread0_inst5_n82_aliasMSR: 2112_9A00h
Init: _ccd2_lthree0_core9_thread1_inst5_n83_aliasMSR: 2112_9B00h
Init: _ccd2_lthree0_core10_thread0_inst5_n84_aliasMSR: 2114_9A00h
Init: _ccd2_lthree0_core10_thread1_inst5_n85_aliasMSR: 2114_9B00h
Init: _ccd2_lthree0_core11_thread0_inst5_n86_aliasMSR: 2116_9A00h
Init: _ccd2_lthree0_core11_thread1_inst5_n87_aliasMSR: 2116_9B00h
Init: _ccd2_lthree0_core12_thread0_inst5_n88_aliasMSR: 2118_9A00h
Init: _ccd2_lthree0_core12_thread1_inst5_n89_aliasMSR: 2118_9B00h
Init: _ccd2_lthree0_core13_thread0_inst5_n90_aliasMSR: 211A_9A00h
Init: _ccd2_lthree0_core13_thread1_inst5_n91_aliasMSR: 211A_9B00h
Init: _ccd2_lthree0_core14_thread0_inst5_n92_aliasMSR: 211C_9A00h
Init: _ccd2_lthree0_core14_thread1_inst5_n93_aliasMSR: 211C_9B00h
Init: _ccd2_lthree0_core15_thread0_inst5_n94_aliasMSR: 211E_9A00h
Init: _ccd2_lthree0_core15_thread1_inst5_n95_aliasMSR: 211E_9B00h
Init: _ccd3_lthree0_core0_thread0_inst5_n96_aliasMSR: 2180_9A00h
Init: _ccd3_lthree0_core0_thread1_inst5_n97_aliasMSR: 2180_9B00h
Init: _ccd3_lthree0_core1_thread0_inst5_n98_aliasMSR: 2182_9A00h
Init: _ccd3_lthree0_core1_thread1_inst5_n99_aliasMSR: 2182_9B00h
Init: _ccd3_lthree0_core2_thread0_inst5_n100_aliasMSR: 2184_9A00h
Init: _ccd3_lthree0_core2_thread1_inst5_n101_aliasMSR: 2184_9B00h
Init: _ccd3_lthree0_core3_thread0_inst5_n102_aliasMSR: 2186_9A00h
Init: _ccd3_lthree0_core3_thread1_inst5_n103_aliasMSR: 2186_9B00h
Init: _ccd3_lthree0_core4_thread0_inst5_n104_aliasMSR: 2188_9A00h
Init: _ccd3_lthree0_core4_thread1_inst5_n105_aliasMSR: 2188_9B00h
Init: _ccd3_lthree0_core5_thread0_inst5_n106_aliasMSR: 218A_9A00h
Init: _ccd3_lthree0_core5_thread1_inst5_n107_aliasMSR: 218A_9B00h
Init: _ccd3_lthree0_core6_thread0_inst5_n108_aliasMSR: 218C_9A00h
Init: _ccd3_lthree0_core6_thread1_inst5_n109_aliasMSR: 218C_9B00h
Init: _ccd3_lthree0_core7_thread0_inst5_n110_aliasMSR: 218E_9A00h
Init: _ccd3_lthree0_core7_thread1_inst5_n111_aliasMSR: 218E_9B00h
Init: _ccd3_lthree0_core8_thread0_inst5_n112_aliasMSR: 2190_9A00h
Init: _ccd3_lthree0_core8_thread1_inst5_n113_aliasMSR: 2190_9B00h

Init: _ccd3_lthree0_core9_thread0_inst5_n114_aliasMSR: 2192_9A00h
Init: _ccd3_lthree0_core9_thread1_inst5_n115_aliasMSR: 2192_9B00h
Init: _ccd3_lthree0_core10_thread0_inst5_n116_aliasMSR: 2194_9A00h
Init: _ccd3_lthree0_core10_thread1_inst5_n117_aliasMSR: 2194_9B00h
Init: _ccd3_lthree0_core11_thread0_inst5_n118_aliasMSR: 2196_9A00h
Init: _ccd3_lthree0_core11_thread1_inst5_n119_aliasMSR: 2196_9B00h
Init: _ccd3_lthree0_core12_thread0_inst5_n120_aliasMSR: 2198_9A00h
Init: _ccd3_lthree0_core12_thread1_inst5_n121_aliasMSR: 2198_9B00h
Init: _ccd3_lthree0_core13_thread0_inst5_n122_aliasMSR: 219A_9A00h
Init: _ccd3_lthree0_core13_thread1_inst5_n123_aliasMSR: 219A_9B00h
Init: _ccd3_lthree0_core14_thread0_inst5_n124_aliasMSR: 219C_9A00h
Init: _ccd3_lthree0_core14_thread1_inst5_n125_aliasMSR: 219C_9B00h
Init: _ccd3_lthree0_core15_thread0_inst5_n126_aliasMSR: 219E_9A00h
Init: _ccd3_lthree0_core15_thread1_inst5_n127_aliasMSR: 219E_9B00h
Init: _ccd4_lthree0_core0_thread0_inst5_n128_aliasMSR: 2200_9A00h
Init: _ccd4_lthree0_core0_thread1_inst5_n129_aliasMSR: 2200_9B00h
Init: _ccd4_lthree0_core1_thread0_inst5_n130_aliasMSR: 2202_9A00h
Init: _ccd4_lthree0_core1_thread1_inst5_n131_aliasMSR: 2202_9B00h
Init: _ccd4_lthree0_core2_thread0_inst5_n132_aliasMSR: 2204_9A00h
Init: _ccd4_lthree0_core2_thread1_inst5_n133_aliasMSR: 2204_9B00h
Init: _ccd4_lthree0_core3_thread0_inst5_n134_aliasMSR: 2206_9A00h
Init: _ccd4_lthree0_core3_thread1_inst5_n135_aliasMSR: 2206_9B00h
Init: _ccd4_lthree0_core4_thread0_inst5_n136_aliasMSR: 2208_9A00h
Init: _ccd4_lthree0_core4_thread1_inst5_n137_aliasMSR: 2208_9B00h
Init: _ccd4_lthree0_core5_thread0_inst5_n138_aliasMSR: 220A_9A00h
Init: _ccd4_lthree0_core5_thread1_inst5_n139_aliasMSR: 220A_9B00h
Init: _ccd4_lthree0_core6_thread0_inst5_n140_aliasMSR: 220C_9A00h
Init: _ccd4_lthree0_core6_thread1_inst5_n141_aliasMSR: 220C_9B00h
Init: _ccd4_lthree0_core7_thread0_inst5_n142_aliasMSR: 220E_9A00h
Init: _ccd4_lthree0_core7_thread1_inst5_n143_aliasMSR: 220E_9B00h
Init: _ccd4_lthree0_core8_thread0_inst5_n144_aliasMSR: 2210_9A00h
Init: _ccd4_lthree0_core8_thread1_inst5_n145_aliasMSR: 2210_9B00h
Init: _ccd4_lthree0_core9_thread0_inst5_n146_aliasMSR: 2212_9A00h
Init: _ccd4_lthree0_core9_thread1_inst5_n147_aliasMSR: 2212_9B00h
Init: _ccd4_lthree0_core10_thread0_inst5_n148_aliasMSR: 2214_9A00h
Init: _ccd4_lthree0_core10_thread1_inst5_n149_aliasMSR: 2214_9B00h
Init: _ccd4_lthree0_core11_thread0_inst5_n150_aliasMSR: 2216_9A00h
Init: _ccd4_lthree0_core11_thread1_inst5_n151_aliasMSR: 2216_9B00h
Init: _ccd4_lthree0_core12_thread0_inst5_n152_aliasMSR: 2218_9A00h
Init: _ccd4_lthree0_core12_thread1_inst5_n153_aliasMSR: 2218_9B00h
Init: _ccd4_lthree0_core13_thread0_inst5_n154_aliasMSR: 221A_9A00h
Init: _ccd4_lthree0_core13_thread1_inst5_n155_aliasMSR: 221A_9B00h
Init: _ccd4_lthree0_core14_thread0_inst5_n156_aliasMSR: 221C_9A00h
Init: _ccd4_lthree0_core14_thread1_inst5_n157_aliasMSR: 221C_9B00h
Init: _ccd4_lthree0_core15_thread0_inst5_n158_aliasMSR: 221E_9A00h
Init: _ccd4_lthree0_core15_thread1_inst5_n159_aliasMSR: 221E_9B00h
Init: _ccd5_lthree0_core0_thread0_inst5_n160_aliasMSR: 2280_9A00h
Init: _ccd5_lthree0_core0_thread1_inst5_n161_aliasMSR: 2280_9B00h
Init: _ccd5_lthree0_core1_thread0_inst5_n162_aliasMSR: 2282_9A00h
Init: _ccd5_lthree0_core1_thread1_inst5_n163_aliasMSR: 2282_9B00h
Init: _ccd5_lthree0_core2_thread0_inst5_n164_aliasMSR: 2284_9A00h
Init: _ccd5_lthree0_core2_thread1_inst5_n165_aliasMSR: 2284_9B00h
Init: _ccd5_lthree0_core3_thread0_inst5_n166_aliasMSR: 2286_9A00h
Init: _ccd5_lthree0_core3_thread1_inst5_n167_aliasMSR: 2286_9B00h
Init: _ccd5_lthree0_core4_thread0_inst5_n168_aliasMSR: 2288_9A00h
Init: _ccd5_lthree0_core4_thread1_inst5_n169_aliasMSR: 2288_9B00h
Init: _ccd5_lthree0_core5_thread0_inst5_n170_aliasMSR: 228A_9A00h
Init: _ccd5_lthree0_core5_thread1_inst5_n171_aliasMSR: 228A_9B00h
Init: _ccd5_lthree0_core6_thread0_inst5_n172_aliasMSR: 228C_9A00h
Init: _ccd5_lthree0_core6_thread1_inst5_n173_aliasMSR: 228C_9B00h
Init: _ccd5_lthree0_core7_thread0_inst5_n174_aliasMSR: 228E_9A00h
Init: _ccd5_lthree0_core7_thread1_inst5_n175_aliasMSR: 228E_9B00h
Init: _ccd5_lthree0_core8_thread0_inst5_n176_aliasMSR: 2290_9A00h
Init: _ccd5_lthree0_core8_thread1_inst5_n177_aliasMSR: 2290_9B00h
Init: _ccd5_lthree0_core9_thread0_inst5_n178_aliasMSR: 2292_9A00h

Init: _ccd5_lthree0_core9_thread1_inst5_n179_aliasMSR: 2292_9B00h
Init: _ccd5_lthree0_core10_thread0_inst5_n180_aliasMSR: 2294_9A00h
Init: _ccd5_lthree0_core10_thread1_inst5_n181_aliasMSR: 2294_9B00h
Init: _ccd5_lthree0_core11_thread0_inst5_n182_aliasMSR: 2296_9A00h
Init: _ccd5_lthree0_core11_thread1_inst5_n183_aliasMSR: 2296_9B00h
Init: _ccd5_lthree0_core12_thread0_inst5_n184_aliasMSR: 2298_9A00h
Init: _ccd5_lthree0_core12_thread1_inst5_n185_aliasMSR: 2298_9B00h
Init: _ccd5_lthree0_core13_thread0_inst5_n186_aliasMSR: 229A_9A00h
Init: _ccd5_lthree0_core13_thread1_inst5_n187_aliasMSR: 229A_9B00h
Init: _ccd5_lthree0_core14_thread0_inst5_n188_aliasMSR: 229C_9A00h
Init: _ccd5_lthree0_core14_thread1_inst5_n189_aliasMSR: 229C_9B00h
Init: _ccd5_lthree0_core15_thread0_inst5_n190_aliasMSR: 229E_9A00h
Init: _ccd5_lthree0_core15_thread1_inst5_n191_aliasMSR: 229E_9B00h
Init: _ccd6_lthree0_core0_thread0_inst5_n192_aliasMSR: 2300_9A00h
Init: _ccd6_lthree0_core0_thread1_inst5_n193_aliasMSR: 2300_9B00h
Init: _ccd6_lthree0_core1_thread0_inst5_n194_aliasMSR: 2302_9A00h
Init: _ccd6_lthree0_core1_thread1_inst5_n195_aliasMSR: 2302_9B00h
Init: _ccd6_lthree0_core2_thread0_inst5_n196_aliasMSR: 2304_9A00h
Init: _ccd6_lthree0_core2_thread1_inst5_n197_aliasMSR: 2304_9B00h
Init: _ccd6_lthree0_core3_thread0_inst5_n198_aliasMSR: 2306_9A00h
Init: _ccd6_lthree0_core3_thread1_inst5_n199_aliasMSR: 2306_9B00h
Init: _ccd6_lthree0_core4_thread0_inst5_n200_aliasMSR: 2308_9A00h
Init: _ccd6_lthree0_core4_thread1_inst5_n201_aliasMSR: 2308_9B00h
Init: _ccd6_lthree0_core5_thread0_inst5_n202_aliasMSR: 230A_9A00h
Init: _ccd6_lthree0_core5_thread1_inst5_n203_aliasMSR: 230A_9B00h
Init: _ccd6_lthree0_core6_thread0_inst5_n204_aliasMSR: 230C_9A00h
Init: _ccd6_lthree0_core6_thread1_inst5_n205_aliasMSR: 230C_9B00h
Init: _ccd6_lthree0_core7_thread0_inst5_n206_aliasMSR: 230E_9A00h
Init: _ccd6_lthree0_core7_thread1_inst5_n207_aliasMSR: 230E_9B00h
Init: _ccd6_lthree0_core8_thread0_inst5_n208_aliasMSR: 2310_9A00h
Init: _ccd6_lthree0_core8_thread1_inst5_n209_aliasMSR: 2310_9B00h
Init: _ccd6_lthree0_core9_thread0_inst5_n210_aliasMSR: 2312_9A00h
Init: _ccd6_lthree0_core9_thread1_inst5_n211_aliasMSR: 2312_9B00h
Init: _ccd6_lthree0_core10_thread0_inst5_n212_aliasMSR: 2314_9A00h
Init: _ccd6_lthree0_core10_thread1_inst5_n213_aliasMSR: 2314_9B00h
Init: _ccd6_lthree0_core11_thread0_inst5_n214_aliasMSR: 2316_9A00h
Init: _ccd6_lthree0_core11_thread1_inst5_n215_aliasMSR: 2316_9B00h
Init: _ccd6_lthree0_core12_thread0_inst5_n216_aliasMSR: 2318_9A00h
Init: _ccd6_lthree0_core12_thread1_inst5_n217_aliasMSR: 2318_9B00h
Init: _ccd6_lthree0_core13_thread0_inst5_n218_aliasMSR: 231A_9A00h
Init: _ccd6_lthree0_core13_thread1_inst5_n219_aliasMSR: 231A_9B00h
Init: _ccd6_lthree0_core14_thread0_inst5_n220_aliasMSR: 231C_9A00h
Init: _ccd6_lthree0_core14_thread1_inst5_n221_aliasMSR: 231C_9B00h
Init: _ccd6_lthree0_core15_thread0_inst5_n222_aliasMSR: 231E_9A00h
Init: _ccd6_lthree0_core15_thread1_inst5_n223_aliasMSR: 231E_9B00h
Init: _ccd7_lthree0_core0_thread0_inst5_n224_aliasMSR: 2380_9A00h
Init: _ccd7_lthree0_core0_thread1_inst5_n225_aliasMSR: 2380_9B00h
Init: _ccd7_lthree0_core1_thread0_inst5_n226_aliasMSR: 2382_9A00h
Init: _ccd7_lthree0_core1_thread1_inst5_n227_aliasMSR: 2382_9B00h
Init: _ccd7_lthree0_core2_thread0_inst5_n228_aliasMSR: 2384_9A00h
Init: _ccd7_lthree0_core2_thread1_inst5_n229_aliasMSR: 2384_9B00h
Init: _ccd7_lthree0_core3_thread0_inst5_n230_aliasMSR: 2386_9A00h
Init: _ccd7_lthree0_core3_thread1_inst5_n231_aliasMSR: 2386_9B00h
Init: _ccd7_lthree0_core4_thread0_inst5_n232_aliasMSR: 2388_9A00h
Init: _ccd7_lthree0_core4_thread1_inst5_n233_aliasMSR: 2388_9B00h
Init: _ccd7_lthree0_core5_thread0_inst5_n234_aliasMSR: 238A_9A00h
Init: _ccd7_lthree0_core5_thread1_inst5_n235_aliasMSR: 238A_9B00h
Init: _ccd7_lthree0_core6_thread0_inst5_n236_aliasMSR: 238C_9A00h
Init: _ccd7_lthree0_core6_thread1_inst5_n237_aliasMSR: 238C_9B00h
Init: _ccd7_lthree0_core7_thread0_inst5_n238_aliasMSR: 238E_9A00h
Init: _ccd7_lthree0_core7_thread1_inst5_n239_aliasMSR: 238E_9B00h
Init: _ccd7_lthree0_core8_thread0_inst5_n240_aliasMSR: 2390_9A00h
Init: _ccd7_lthree0_core8_thread1_inst5_n241_aliasMSR: 2390_9B00h
Init: _ccd7_lthree0_core9_thread0_inst5_n242_aliasMSR: 2392_9A00h
Init: _ccd7_lthree0_core9_thread1_inst5_n243_aliasMSR: 2392_9B00h

Init: _ccd7_lthree0_core10_thread0_inst5_n244_aliasMSR: 2394_9A00h
Init: _ccd7_lthree0_core10_thread1_inst5_n245_aliasMSR: 2394_9B00h
Init: _ccd7_lthree0_core11_thread0_inst5_n246_aliasMSR: 2396_9A00h
Init: _ccd7_lthree0_core11_thread1_inst5_n247_aliasMSR: 2396_9B00h
Init: _ccd7_lthree0_core12_thread0_inst5_n248_aliasMSR: 2398_9A00h
Init: _ccd7_lthree0_core12_thread1_inst5_n249_aliasMSR: 2398_9B00h
Init: _ccd7_lthree0_core13_thread0_inst5_n250_aliasMSR: 239A_9A00h
Init: _ccd7_lthree0_core13_thread1_inst5_n251_aliasMSR: 239A_9B00h
Init: _ccd7_lthree0_core14_thread0_inst5_n252_aliasMSR: 239C_9A00h
Init: _ccd7_lthree0_core14_thread1_inst5_n253_aliasMSR: 239C_9B00h
Init: _ccd7_lthree0_core15_thread0_inst5_n254_aliasMSR: 239E_9A00h
Init: _ccd7_lthree0_core15_thread1_inst5_n255_aliasMSR: 239E_9B00h
Init: _ccd8_lthree0_core0_thread0_inst5_n256_aliasMSR: 2400_9A00h
Init: _ccd8_lthree0_core0_thread1_inst5_n257_aliasMSR: 2400_9B00h
Init: _ccd8_lthree0_core1_thread0_inst5_n258_aliasMSR: 2402_9A00h
Init: _ccd8_lthree0_core1_thread1_inst5_n259_aliasMSR: 2402_9B00h
Init: _ccd8_lthree0_core2_thread0_inst5_n260_aliasMSR: 2404_9A00h
Init: _ccd8_lthree0_core2_thread1_inst5_n261_aliasMSR: 2404_9B00h
Init: _ccd8_lthree0_core3_thread0_inst5_n262_aliasMSR: 2406_9A00h
Init: _ccd8_lthree0_core3_thread1_inst5_n263_aliasMSR: 2406_9B00h
Init: _ccd8_lthree0_core4_thread0_inst5_n264_aliasMSR: 2408_9A00h
Init: _ccd8_lthree0_core4_thread1_inst5_n265_aliasMSR: 2408_9B00h
Init: _ccd8_lthree0_core5_thread0_inst5_n266_aliasMSR: 240A_9A00h
Init: _ccd8_lthree0_core5_thread1_inst5_n267_aliasMSR: 240A_9B00h
Init: _ccd8_lthree0_core6_thread0_inst5_n268_aliasMSR: 240C_9A00h
Init: _ccd8_lthree0_core6_thread1_inst5_n269_aliasMSR: 240C_9B00h
Init: _ccd8_lthree0_core7_thread0_inst5_n270_aliasMSR: 240E_9A00h
Init: _ccd8_lthree0_core7_thread1_inst5_n271_aliasMSR: 240E_9B00h
Init: _ccd8_lthree0_core8_thread0_inst5_n272_aliasMSR: 2410_9A00h
Init: _ccd8_lthree0_core8_thread1_inst5_n273_aliasMSR: 2410_9B00h
Init: _ccd8_lthree0_core9_thread0_inst5_n274_aliasMSR: 2412_9A00h
Init: _ccd8_lthree0_core9_thread1_inst5_n275_aliasMSR: 2412_9B00h
Init: _ccd8_lthree0_core10_thread0_inst5_n276_aliasMSR: 2414_9A00h
Init: _ccd8_lthree0_core10_thread1_inst5_n277_aliasMSR: 2414_9B00h
Init: _ccd8_lthree0_core11_thread0_inst5_n278_aliasMSR: 2416_9A00h
Init: _ccd8_lthree0_core11_thread1_inst5_n279_aliasMSR: 2416_9B00h
Init: _ccd8_lthree0_core12_thread0_inst5_n280_aliasMSR: 2418_9A00h
Init: _ccd8_lthree0_core12_thread1_inst5_n281_aliasMSR: 2418_9B00h
Init: _ccd8_lthree0_core13_thread0_inst5_n282_aliasMSR: 241A_9A00h
Init: _ccd8_lthree0_core13_thread1_inst5_n283_aliasMSR: 241A_9B00h
Init: _ccd8_lthree0_core14_thread0_inst5_n284_aliasMSR: 241C_9A00h
Init: _ccd8_lthree0_core14_thread1_inst5_n285_aliasMSR: 241C_9B00h
Init: _ccd8_lthree0_core15_thread0_inst5_n286_aliasMSR: 241E_9A00h
Init: _ccd8_lthree0_core15_thread1_inst5_n287_aliasMSR: 241E_9B00h
Init: _ccd9_lthree0_core0_thread0_inst5_n288_aliasMSR: 2480_9A00h
Init: _ccd9_lthree0_core0_thread1_inst5_n289_aliasMSR: 2480_9B00h
Init: _ccd9_lthree0_core1_thread0_inst5_n290_aliasMSR: 2482_9A00h
Init: _ccd9_lthree0_core1_thread1_inst5_n291_aliasMSR: 2482_9B00h
Init: _ccd9_lthree0_core2_thread0_inst5_n292_aliasMSR: 2484_9A00h
Init: _ccd9_lthree0_core2_thread1_inst5_n293_aliasMSR: 2484_9B00h
Init: _ccd9_lthree0_core3_thread0_inst5_n294_aliasMSR: 2486_9A00h
Init: _ccd9_lthree0_core3_thread1_inst5_n295_aliasMSR: 2486_9B00h
Init: _ccd9_lthree0_core4_thread0_inst5_n296_aliasMSR: 2488_9A00h
Init: _ccd9_lthree0_core4_thread1_inst5_n297_aliasMSR: 2488_9B00h
Init: _ccd9_lthree0_core5_thread0_inst5_n298_aliasMSR: 248A_9A00h
Init: _ccd9_lthree0_core5_thread1_inst5_n299_aliasMSR: 248A_9B00h
Init: _ccd9_lthree0_core6_thread0_inst5_n300_aliasMSR: 248C_9A00h
Init: _ccd9_lthree0_core6_thread1_inst5_n301_aliasMSR: 248C_9B00h
Init: _ccd9_lthree0_core7_thread0_inst5_n302_aliasMSR: 248E_9A00h
Init: _ccd9_lthree0_core7_thread1_inst5_n303_aliasMSR: 248E_9B00h
Init: _ccd9_lthree0_core8_thread0_inst5_n304_aliasMSR: 2490_9A00h
Init: _ccd9_lthree0_core8_thread1_inst5_n305_aliasMSR: 2490_9B00h
Init: _ccd9_lthree0_core9_thread0_inst5_n306_aliasMSR: 2492_9A00h
Init: _ccd9_lthree0_core9_thread1_inst5_n307_aliasMSR: 2492_9B00h
Init: _ccd9_lthree0_core10_thread0_inst5_n308_aliasMSR: 2494_9A00h

Init: _ccd9_lthree0_core10_thread1_inst5_n309_aliasMSR: 2494_9B00h
Init: _ccd9_lthree0_core11_thread0_inst5_n310_aliasMSR: 2496_9A00h
Init: _ccd9_lthree0_core11_thread1_inst5_n311_aliasMSR: 2496_9B00h
Init: _ccd9_lthree0_core12_thread0_inst5_n312_aliasMSR: 2498_9A00h
Init: _ccd9_lthree0_core12_thread1_inst5_n313_aliasMSR: 2498_9B00h
Init: _ccd9_lthree0_core13_thread0_inst5_n314_aliasMSR: 249A_9A00h
Init: _ccd9_lthree0_core13_thread1_inst5_n315_aliasMSR: 249A_9B00h
Init: _ccd9_lthree0_core14_thread0_inst5_n316_aliasMSR: 249C_9A00h
Init: _ccd9_lthree0_core14_thread1_inst5_n317_aliasMSR: 249C_9B00h
Init: _ccd9_lthree0_core15_thread0_inst5_n318_aliasMSR: 249E_9A00h
Init: _ccd9_lthree0_core15_thread1_inst5_n319_aliasMSR: 249E_9B00h
Init: _ccd10_lthree0_core0_thread0_inst5_n320_aliasMSR: 2500_9A00h
Init: _ccd10_lthree0_core0_thread1_inst5_n321_aliasMSR: 2500_9B00h
Init: _ccd10_lthree0_core1_thread0_inst5_n322_aliasMSR: 2502_9A00h
Init: _ccd10_lthree0_core1_thread1_inst5_n323_aliasMSR: 2502_9B00h
Init: _ccd10_lthree0_core2_thread0_inst5_n324_aliasMSR: 2504_9A00h
Init: _ccd10_lthree0_core2_thread1_inst5_n325_aliasMSR: 2504_9B00h
Init: _ccd10_lthree0_core3_thread0_inst5_n326_aliasMSR: 2506_9A00h
Init: _ccd10_lthree0_core3_thread1_inst5_n327_aliasMSR: 2506_9B00h
Init: _ccd10_lthree0_core4_thread0_inst5_n328_aliasMSR: 2508_9A00h
Init: _ccd10_lthree0_core4_thread1_inst5_n329_aliasMSR: 2508_9B00h
Init: _ccd10_lthree0_core5_thread0_inst5_n330_aliasMSR: 250A_9A00h
Init: _ccd10_lthree0_core5_thread1_inst5_n331_aliasMSR: 250A_9B00h
Init: _ccd10_lthree0_core6_thread0_inst5_n332_aliasMSR: 250C_9A00h
Init: _ccd10_lthree0_core6_thread1_inst5_n333_aliasMSR: 250C_9B00h
Init: _ccd10_lthree0_core7_thread0_inst5_n334_aliasMSR: 250E_9A00h
Init: _ccd10_lthree0_core7_thread1_inst5_n335_aliasMSR: 250E_9B00h
Init: _ccd10_lthree0_core8_thread0_inst5_n336_aliasMSR: 2510_9A00h
Init: _ccd10_lthree0_core8_thread1_inst5_n337_aliasMSR: 2510_9B00h
Init: _ccd10_lthree0_core9_thread0_inst5_n338_aliasMSR: 2512_9A00h
Init: _ccd10_lthree0_core9_thread1_inst5_n339_aliasMSR: 2512_9B00h
Init: _ccd10_lthree0_core10_thread0_inst5_n340_aliasMSR: 2514_9A00h
Init: _ccd10_lthree0_core10_thread1_inst5_n341_aliasMSR: 2514_9B00h
Init: _ccd10_lthree0_core11_thread0_inst5_n342_aliasMSR: 2516_9A00h
Init: _ccd10_lthree0_core11_thread1_inst5_n343_aliasMSR: 2516_9B00h
Init: _ccd10_lthree0_core12_thread0_inst5_n344_aliasMSR: 2518_9A00h
Init: _ccd10_lthree0_core12_thread1_inst5_n345_aliasMSR: 2518_9B00h
Init: _ccd10_lthree0_core13_thread0_inst5_n346_aliasMSR: 251A_9A00h
Init: _ccd10_lthree0_core13_thread1_inst5_n347_aliasMSR: 251A_9B00h
Init: _ccd10_lthree0_core14_thread0_inst5_n348_aliasMSR: 251C_9A00h
Init: _ccd10_lthree0_core14_thread1_inst5_n349_aliasMSR: 251C_9B00h
Init: _ccd10_lthree0_core15_thread0_inst5_n350_aliasMSR: 251E_9A00h
Init: _ccd10_lthree0_core15_thread1_inst5_n351_aliasMSR: 251E_9B00h
Init: _ccd11_lthree0_core0_thread0_inst5_n352_aliasMSR: 2580_9A00h
Init: _ccd11_lthree0_core0_thread1_inst5_n353_aliasMSR: 2580_9B00h
Init: _ccd11_lthree0_core1_thread0_inst5_n354_aliasMSR: 2582_9A00h
Init: _ccd11_lthree0_core1_thread1_inst5_n355_aliasMSR: 2582_9B00h
Init: _ccd11_lthree0_core2_thread0_inst5_n356_aliasMSR: 2584_9A00h
Init: _ccd11_lthree0_core2_thread1_inst5_n357_aliasMSR: 2584_9B00h
Init: _ccd11_lthree0_core3_thread0_inst5_n358_aliasMSR: 2586_9A00h
Init: _ccd11_lthree0_core3_thread1_inst5_n359_aliasMSR: 2586_9B00h
Init: _ccd11_lthree0_core4_thread0_inst5_n360_aliasMSR: 2588_9A00h
Init: _ccd11_lthree0_core4_thread1_inst5_n361_aliasMSR: 2588_9B00h
Init: _ccd11_lthree0_core5_thread0_inst5_n362_aliasMSR: 258A_9A00h
Init: _ccd11_lthree0_core5_thread1_inst5_n363_aliasMSR: 258A_9B00h
Init: _ccd11_lthree0_core6_thread0_inst5_n364_aliasMSR: 258C_9A00h
Init: _ccd11_lthree0_core6_thread1_inst5_n365_aliasMSR: 258C_9B00h
Init: _ccd11_lthree0_core7_thread0_inst5_n366_aliasMSR: 258E_9A00h
Init: _ccd11_lthree0_core7_thread1_inst5_n367_aliasMSR: 258E_9B00h
Init: _ccd11_lthree0_core8_thread0_inst5_n368_aliasMSR: 2590_9A00h
Init: _ccd11_lthree0_core8_thread1_inst5_n369_aliasMSR: 2590_9B00h
Init: _ccd11_lthree0_core9_thread0_inst5_n370_aliasMSR: 2592_9A00h
Init: _ccd11_lthree0_core9_thread1_inst5_n371_aliasMSR: 2592_9B00h
Init: _ccd11_lthree0_core10_thread0_inst5_n372_aliasMSR: 2594_9A00h
Init: _ccd11_lthree0_core10_thread1_inst5_n373_aliasMSR: 2594_9B00h

Init: _ccd11_lthree0_core11_thread0_inst5_n374_aliasMSR: 2596_9A00h
Init: _ccd11_lthree0_core11_thread1_inst5_n375_aliasMSR: 2596_9B00h
Init: _ccd11_lthree0_core12_thread0_inst5_n376_aliasMSR: 2598_9A00h
Init: _ccd11_lthree0_core12_thread1_inst5_n377_aliasMSR: 2598_9B00h
Init: _ccd11_lthree0_core13_thread0_inst5_n378_aliasMSR: 259A_9A00h
Init: _ccd11_lthree0_core13_thread1_inst5_n379_aliasMSR: 259A_9B00h
Init: _ccd11_lthree0_core14_thread0_inst5_n380_aliasMSR: 259C_9A00h
Init: _ccd11_lthree0_core14_thread1_inst5_n381_aliasMSR: 259C_9B00h
Init: _ccd11_lthree0_core15_thread0_inst5_n382_aliasMSR: 259E_9A00h
Init: _ccd11_lthree0_core15_thread1_inst5_n383_aliasMSR: 259E_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](#)

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst5_n[383:0]_aliasMSR; MSRC000_2056

Bits	Description
63:32	Syndrom . 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	ErrorPriority . 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	Length . Read-write, Volatile . Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::EX::MCA_SYND_EX[Syndrome]. Length values greater than 32 (decimal) are interpreted as equal to 32 (decimal). 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	ErrorInformation . 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_EX].

Table 66: 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]	0

	[1] [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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst5_n[383:0]_aliasMSR; MSRC001_0405

Bits	Description
63:14	Reserved.
13	RETMAP . Read-write. Reset: 0. Retire Map parity error.
12	SPECMAP . Read-write. Reset: 0. PRN/FRN freelist parity error.
11	HWA . Read-write. Reset: 0. Hardware Assertion error.
10	BBQ . Read-write. Reset: 0. Branch buffer queue parity error.
9	SQ . Read-write. Reset: 0. EXTID parity error.
8	STATQ . Read-write. Reset: 0. Retire status queue parity error.
7	RETDISP . Read-write. Reset: 0. Retire dispatch queue parity error.
6	CHKPTQ . Read-write. Reset: 0. CHKPTQ. Checkpoint queue parity error.
5	PLDAL . Read-write. Reset: 0. EX payload parity error.
4	PLDAG . Read-write. Reset: 0. Address generator payload parity error.
3	IDRF . Read-write. Reset: 0. Immediate displacement register file parity error.
2	FRF . Read-write. Reset: 0. Flag register file parity error.
1	PRF . Read-write. Reset: 0. Physical register file parity error.
0	WDT . 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](#)

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst5_n[383:0]_aliasMSR; MSRC000_205E

Bits	Description
63:0	Syndrom . 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](#)

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst5_n[383:0]_aliasMSR; MSRC000_205F

Bits	Description
63:0	Syndrom . 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst6_n[383:0]_aliasMSRLEGACY; MSR0000_0418

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst6_n[383:0]_aliasMSR; MSRC000_2060

Bits	Description
63:8	Reserved.
7	KRF . Read-write. Reset: 0. Physical K mask register file (KRF) parity error.
6	HWA . Read-write. Reset: 0. Hardware assertion.
5	SRF . Read-write. Reset: 0. Status register file (SRF) parity error.
4	RQ . Read-write. Reset: 0. Retire queue (RQ) parity error.
3	NSQ . Read-write. Reset: 0. NSQ parity error.
2	SCH . Read-write. Reset: 0. Schedule queue parity error.
1	FL . Read-write. Reset: 0. Freelist (FL) parity error.
0	PRF . 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst6_n[383:0]_aliasMSRLEGACY; MSR0000_0419

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst6_n[383:0]_aliasMSR; MSRC000_2061

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. 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	MiscV. 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	AddrV. 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	PCC. 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	ErrCoreIdVal. 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	TCC. 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	RESERV54. 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	SyndV. 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	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. 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	RESERV22. 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	ErrorCodeExt. 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	ErrorCode. 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_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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst6_n[383:0]_aliasMSRLEGACY; MSR0000_041A

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst6_n[383:0]_aliasMSR; MSRC000_2062

Bits	Description
63:0	ErrorAddr. 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 68: 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst6_n[383:0]_aliasMSRLEGACY; MSR0000_041B

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst6_n[383:0]_aliasMSR; MSRC000_2063

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr:: HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr:: HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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	LvtOffset. 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	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr:: HWCR[McStatusWrEn] !MCA::FP::MCA_MISC0_FP[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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::FP::MCA_MISC0_FP[Locked]) ? Read-write : Read-only.
48	Ovrflw. 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	ErrCnt. 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	BlkPtr. 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst6_n[383:0]_aliasMSR; MSRC000_2064

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	McaXEnable. 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	IntPresent. 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	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::FP::MCA_CONFIG_FP[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. 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	DeferredErrorLoggingSupported. 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	McaX. 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst6_n[383:0]_aliasMSR; MSRC000_2065

Bits	Description
63:48	McaType . Read-only. Reset: 0006h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi . 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	HardwareID . Read-only. Reset: 0B0h. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _ccd0_lthree0_core0_thread0_inst6_n0_aliasMSR: 2000_8600h
	Init: _ccd0_lthree0_core0_thread1_inst6_n1_aliasMSR: 2000_8700h
	Init: _ccd0_lthree0_core1_thread0_inst6_n2_aliasMSR: 2002_8600h
	Init: _ccd0_lthree0_core1_thread1_inst6_n3_aliasMSR: 2002_8700h
	Init: _ccd0_lthree0_core2_thread0_inst6_n4_aliasMSR: 2004_8600h
	Init: _ccd0_lthree0_core2_thread1_inst6_n5_aliasMSR: 2004_8700h
	Init: _ccd0_lthree0_core3_thread0_inst6_n6_aliasMSR: 2006_8600h
	Init: _ccd0_lthree0_core3_thread1_inst6_n7_aliasMSR: 2006_8700h
	Init: _ccd0_lthree0_core4_thread0_inst6_n8_aliasMSR: 2008_8600h
	Init: _ccd0_lthree0_core4_thread1_inst6_n9_aliasMSR: 2008_8700h
	Init: _ccd0_lthree0_core5_thread0_inst6_n10_aliasMSR: 200A_8600h
	Init: _ccd0_lthree0_core5_thread1_inst6_n11_aliasMSR: 200A_8700h
	Init: _ccd0_lthree0_core6_thread0_inst6_n12_aliasMSR: 200C_8600h
	Init: _ccd0_lthree0_core6_thread1_inst6_n13_aliasMSR: 200C_8700h
	Init: _ccd0_lthree0_core7_thread0_inst6_n14_aliasMSR: 200E_8600h
	Init: _ccd0_lthree0_core7_thread1_inst6_n15_aliasMSR: 200E_8700h
	Init: _ccd0_lthree0_core8_thread0_inst6_n16_aliasMSR: 2010_8600h
	Init: _ccd0_lthree0_core8_thread1_inst6_n17_aliasMSR: 2010_8700h
	Init: _ccd0_lthree0_core9_thread0_inst6_n18_aliasMSR: 2012_8600h
	Init: _ccd0_lthree0_core9_thread1_inst6_n19_aliasMSR: 2012_8700h
	Init: _ccd0_lthree0_core10_thread0_inst6_n20_aliasMSR: 2014_8600h
	Init: _ccd0_lthree0_core10_thread1_inst6_n21_aliasMSR: 2014_8700h
	Init: _ccd0_lthree0_core11_thread0_inst6_n22_aliasMSR: 2016_8600h
	Init: _ccd0_lthree0_core11_thread1_inst6_n23_aliasMSR: 2016_8700h
	Init: _ccd0_lthree0_core12_thread0_inst6_n24_aliasMSR: 2018_8600h
	Init: _ccd0_lthree0_core12_thread1_inst6_n25_aliasMSR: 2018_8700h
	Init: _ccd0_lthree0_core13_thread0_inst6_n26_aliasMSR: 201A_8600h
	Init: _ccd0_lthree0_core13_thread1_inst6_n27_aliasMSR: 201A_8700h
	Init: _ccd0_lthree0_core14_thread0_inst6_n28_aliasMSR: 201C_8600h
	Init: _ccd0_lthree0_core14_thread1_inst6_n29_aliasMSR: 201C_8700h
	Init: _ccd0_lthree0_core15_thread0_inst6_n30_aliasMSR: 201E_8600h
	Init: _ccd0_lthree0_core15_thread1_inst6_n31_aliasMSR: 201E_8700h
	Init: _ccd1_lthree0_core0_thread0_inst6_n32_aliasMSR: 2080_8600h
	Init: _ccd1_lthree0_core0_thread1_inst6_n33_aliasMSR: 2080_8700h
	Init: _ccd1_lthree0_core1_thread0_inst6_n34_aliasMSR: 2082_8600h
	Init: _ccd1_lthree0_core1_thread1_inst6_n35_aliasMSR: 2082_8700h
	Init: _ccd1_lthree0_core2_thread0_inst6_n36_aliasMSR: 2084_8600h
	Init: _ccd1_lthree0_core2_thread1_inst6_n37_aliasMSR: 2084_8700h
	Init: _ccd1_lthree0_core3_thread0_inst6_n38_aliasMSR: 2086_8600h
	Init: _ccd1_lthree0_core3_thread1_inst6_n39_aliasMSR: 2086_8700h
	Init: _ccd1_lthree0_core4_thread0_inst6_n40_aliasMSR: 2088_8600h
	Init: _ccd1_lthree0_core4_thread1_inst6_n41_aliasMSR: 2088_8700h
	Init: _ccd1_lthree0_core5_thread0_inst6_n42_aliasMSR: 208A_8600h
	Init: _ccd1_lthree0_core5_thread1_inst6_n43_aliasMSR: 208A_8700h
	Init: _ccd1_lthree0_core6_thread0_inst6_n44_aliasMSR: 208C_8600h
	Init: _ccd1_lthree0_core6_thread1_inst6_n45_aliasMSR: 208C_8700h
	Init: _ccd1_lthree0_core7_thread0_inst6_n46_aliasMSR: 208E_8600h
	Init: _ccd1_lthree0_core7_thread1_inst6_n47_aliasMSR: 208E_8700h
	Init: _ccd1_lthree0_core8_thread0_inst6_n48_aliasMSR: 2090_8600h

Init: _ccd1_lthree0_core8_thread1_inst6_n49_aliasMSR: 2090_8700h
Init: _ccd1_lthree0_core9_thread0_inst6_n50_aliasMSR: 2092_8600h
Init: _ccd1_lthree0_core9_thread1_inst6_n51_aliasMSR: 2092_8700h
Init: _ccd1_lthree0_core10_thread0_inst6_n52_aliasMSR: 2094_8600h
Init: _ccd1_lthree0_core10_thread1_inst6_n53_aliasMSR: 2094_8700h
Init: _ccd1_lthree0_core11_thread0_inst6_n54_aliasMSR: 2096_8600h
Init: _ccd1_lthree0_core11_thread1_inst6_n55_aliasMSR: 2096_8700h
Init: _ccd1_lthree0_core12_thread0_inst6_n56_aliasMSR: 2098_8600h
Init: _ccd1_lthree0_core12_thread1_inst6_n57_aliasMSR: 2098_8700h
Init: _ccd1_lthree0_core13_thread0_inst6_n58_aliasMSR: 209A_8600h
Init: _ccd1_lthree0_core13_thread1_inst6_n59_aliasMSR: 209A_8700h
Init: _ccd1_lthree0_core14_thread0_inst6_n60_aliasMSR: 209C_8600h
Init: _ccd1_lthree0_core14_thread1_inst6_n61_aliasMSR: 209C_8700h
Init: _ccd1_lthree0_core15_thread0_inst6_n62_aliasMSR: 209E_8600h
Init: _ccd1_lthree0_core15_thread1_inst6_n63_aliasMSR: 209E_8700h
Init: _ccd2_lthree0_core0_thread0_inst6_n64_aliasMSR: 2100_8600h
Init: _ccd2_lthree0_core0_thread1_inst6_n65_aliasMSR: 2100_8700h
Init: _ccd2_lthree0_core1_thread0_inst6_n66_aliasMSR: 2102_8600h
Init: _ccd2_lthree0_core1_thread1_inst6_n67_aliasMSR: 2102_8700h
Init: _ccd2_lthree0_core2_thread0_inst6_n68_aliasMSR: 2104_8600h
Init: _ccd2_lthree0_core2_thread1_inst6_n69_aliasMSR: 2104_8700h
Init: _ccd2_lthree0_core3_thread0_inst6_n70_aliasMSR: 2106_8600h
Init: _ccd2_lthree0_core3_thread1_inst6_n71_aliasMSR: 2106_8700h
Init: _ccd2_lthree0_core4_thread0_inst6_n72_aliasMSR: 2108_8600h
Init: _ccd2_lthree0_core4_thread1_inst6_n73_aliasMSR: 2108_8700h
Init: _ccd2_lthree0_core5_thread0_inst6_n74_aliasMSR: 210A_8600h
Init: _ccd2_lthree0_core5_thread1_inst6_n75_aliasMSR: 210A_8700h
Init: _ccd2_lthree0_core6_thread0_inst6_n76_aliasMSR: 210C_8600h
Init: _ccd2_lthree0_core6_thread1_inst6_n77_aliasMSR: 210C_8700h
Init: _ccd2_lthree0_core7_thread0_inst6_n78_aliasMSR: 210E_8600h
Init: _ccd2_lthree0_core7_thread1_inst6_n79_aliasMSR: 210E_8700h
Init: _ccd2_lthree0_core8_thread0_inst6_n80_aliasMSR: 2110_8600h
Init: _ccd2_lthree0_core8_thread1_inst6_n81_aliasMSR: 2110_8700h
Init: _ccd2_lthree0_core9_thread0_inst6_n82_aliasMSR: 2112_8600h
Init: _ccd2_lthree0_core9_thread1_inst6_n83_aliasMSR: 2112_8700h
Init: _ccd2_lthree0_core10_thread0_inst6_n84_aliasMSR: 2114_8600h
Init: _ccd2_lthree0_core10_thread1_inst6_n85_aliasMSR: 2114_8700h
Init: _ccd2_lthree0_core11_thread0_inst6_n86_aliasMSR: 2116_8600h
Init: _ccd2_lthree0_core11_thread1_inst6_n87_aliasMSR: 2116_8700h
Init: _ccd2_lthree0_core12_thread0_inst6_n88_aliasMSR: 2118_8600h
Init: _ccd2_lthree0_core12_thread1_inst6_n89_aliasMSR: 2118_8700h
Init: _ccd2_lthree0_core13_thread0_inst6_n90_aliasMSR: 211A_8600h
Init: _ccd2_lthree0_core13_thread1_inst6_n91_aliasMSR: 211A_8700h
Init: _ccd2_lthree0_core14_thread0_inst6_n92_aliasMSR: 211C_8600h
Init: _ccd2_lthree0_core14_thread1_inst6_n93_aliasMSR: 211C_8700h
Init: _ccd2_lthree0_core15_thread0_inst6_n94_aliasMSR: 211E_8600h
Init: _ccd2_lthree0_core15_thread1_inst6_n95_aliasMSR: 211E_8700h
Init: _ccd3_lthree0_core0_thread0_inst6_n96_aliasMSR: 2180_8600h
Init: _ccd3_lthree0_core0_thread1_inst6_n97_aliasMSR: 2180_8700h
Init: _ccd3_lthree0_core1_thread0_inst6_n98_aliasMSR: 2182_8600h
Init: _ccd3_lthree0_core1_thread1_inst6_n99_aliasMSR: 2182_8700h
Init: _ccd3_lthree0_core2_thread0_inst6_n100_aliasMSR: 2184_8600h
Init: _ccd3_lthree0_core2_thread1_inst6_n101_aliasMSR: 2184_8700h
Init: _ccd3_lthree0_core3_thread0_inst6_n102_aliasMSR: 2186_8600h
Init: _ccd3_lthree0_core3_thread1_inst6_n103_aliasMSR: 2186_8700h
Init: _ccd3_lthree0_core4_thread0_inst6_n104_aliasMSR: 2188_8600h
Init: _ccd3_lthree0_core4_thread1_inst6_n105_aliasMSR: 2188_8700h
Init: _ccd3_lthree0_core5_thread0_inst6_n106_aliasMSR: 218A_8600h
Init: _ccd3_lthree0_core5_thread1_inst6_n107_aliasMSR: 218A_8700h
Init: _ccd3_lthree0_core6_thread0_inst6_n108_aliasMSR: 218C_8600h
Init: _ccd3_lthree0_core6_thread1_inst6_n109_aliasMSR: 218C_8700h
Init: _ccd3_lthree0_core7_thread0_inst6_n110_aliasMSR: 218E_8600h
Init: _ccd3_lthree0_core7_thread1_inst6_n111_aliasMSR: 218E_8700h
Init: _ccd3_lthree0_core8_thread0_inst6_n112_aliasMSR: 2190_8600h
Init: _ccd3_lthree0_core8_thread1_inst6_n113_aliasMSR: 2190_8700h

Init: _ccd3_lthree0_core9_thread0_inst6_n114_aliasMSR: 2192_8600h
Init: _ccd3_lthree0_core9_thread1_inst6_n115_aliasMSR: 2192_8700h
Init: _ccd3_lthree0_core10_thread0_inst6_n116_aliasMSR: 2194_8600h
Init: _ccd3_lthree0_core10_thread1_inst6_n117_aliasMSR: 2194_8700h
Init: _ccd3_lthree0_core11_thread0_inst6_n118_aliasMSR: 2196_8600h
Init: _ccd3_lthree0_core11_thread1_inst6_n119_aliasMSR: 2196_8700h
Init: _ccd3_lthree0_core12_thread0_inst6_n120_aliasMSR: 2198_8600h
Init: _ccd3_lthree0_core12_thread1_inst6_n121_aliasMSR: 2198_8700h
Init: _ccd3_lthree0_core13_thread0_inst6_n122_aliasMSR: 219A_8600h
Init: _ccd3_lthree0_core13_thread1_inst6_n123_aliasMSR: 219A_8700h
Init: _ccd3_lthree0_core14_thread0_inst6_n124_aliasMSR: 219C_8600h
Init: _ccd3_lthree0_core14_thread1_inst6_n125_aliasMSR: 219C_8700h
Init: _ccd3_lthree0_core15_thread0_inst6_n126_aliasMSR: 219E_8600h
Init: _ccd3_lthree0_core15_thread1_inst6_n127_aliasMSR: 219E_8700h
Init: _ccd4_lthree0_core0_thread0_inst6_n128_aliasMSR: 2200_8600h
Init: _ccd4_lthree0_core0_thread1_inst6_n129_aliasMSR: 2200_8700h
Init: _ccd4_lthree0_core1_thread0_inst6_n130_aliasMSR: 2202_8600h
Init: _ccd4_lthree0_core1_thread1_inst6_n131_aliasMSR: 2202_8700h
Init: _ccd4_lthree0_core2_thread0_inst6_n132_aliasMSR: 2204_8600h
Init: _ccd4_lthree0_core2_thread1_inst6_n133_aliasMSR: 2204_8700h
Init: _ccd4_lthree0_core3_thread0_inst6_n134_aliasMSR: 2206_8600h
Init: _ccd4_lthree0_core3_thread1_inst6_n135_aliasMSR: 2206_8700h
Init: _ccd4_lthree0_core4_thread0_inst6_n136_aliasMSR: 2208_8600h
Init: _ccd4_lthree0_core4_thread1_inst6_n137_aliasMSR: 2208_8700h
Init: _ccd4_lthree0_core5_thread0_inst6_n138_aliasMSR: 220A_8600h
Init: _ccd4_lthree0_core5_thread1_inst6_n139_aliasMSR: 220A_8700h
Init: _ccd4_lthree0_core6_thread0_inst6_n140_aliasMSR: 220C_8600h
Init: _ccd4_lthree0_core6_thread1_inst6_n141_aliasMSR: 220C_8700h
Init: _ccd4_lthree0_core7_thread0_inst6_n142_aliasMSR: 220E_8600h
Init: _ccd4_lthree0_core7_thread1_inst6_n143_aliasMSR: 220E_8700h
Init: _ccd4_lthree0_core8_thread0_inst6_n144_aliasMSR: 2210_8600h
Init: _ccd4_lthree0_core8_thread1_inst6_n145_aliasMSR: 2210_8700h
Init: _ccd4_lthree0_core9_thread0_inst6_n146_aliasMSR: 2212_8600h
Init: _ccd4_lthree0_core9_thread1_inst6_n147_aliasMSR: 2212_8700h
Init: _ccd4_lthree0_core10_thread0_inst6_n148_aliasMSR: 2214_8600h
Init: _ccd4_lthree0_core10_thread1_inst6_n149_aliasMSR: 2214_8700h
Init: _ccd4_lthree0_core11_thread0_inst6_n150_aliasMSR: 2216_8600h
Init: _ccd4_lthree0_core11_thread1_inst6_n151_aliasMSR: 2216_8700h
Init: _ccd4_lthree0_core12_thread0_inst6_n152_aliasMSR: 2218_8600h
Init: _ccd4_lthree0_core12_thread1_inst6_n153_aliasMSR: 2218_8700h
Init: _ccd4_lthree0_core13_thread0_inst6_n154_aliasMSR: 221A_8600h
Init: _ccd4_lthree0_core13_thread1_inst6_n155_aliasMSR: 221A_8700h
Init: _ccd4_lthree0_core14_thread0_inst6_n156_aliasMSR: 221C_8600h
Init: _ccd4_lthree0_core14_thread1_inst6_n157_aliasMSR: 221C_8700h
Init: _ccd4_lthree0_core15_thread0_inst6_n158_aliasMSR: 221E_8600h
Init: _ccd4_lthree0_core15_thread1_inst6_n159_aliasMSR: 221E_8700h
Init: _ccd5_lthree0_core0_thread0_inst6_n160_aliasMSR: 2280_8600h
Init: _ccd5_lthree0_core0_thread1_inst6_n161_aliasMSR: 2280_8700h
Init: _ccd5_lthree0_core1_thread0_inst6_n162_aliasMSR: 2282_8600h
Init: _ccd5_lthree0_core1_thread1_inst6_n163_aliasMSR: 2282_8700h
Init: _ccd5_lthree0_core2_thread0_inst6_n164_aliasMSR: 2284_8600h
Init: _ccd5_lthree0_core2_thread1_inst6_n165_aliasMSR: 2284_8700h
Init: _ccd5_lthree0_core3_thread0_inst6_n166_aliasMSR: 2286_8600h
Init: _ccd5_lthree0_core3_thread1_inst6_n167_aliasMSR: 2286_8700h
Init: _ccd5_lthree0_core4_thread0_inst6_n168_aliasMSR: 2288_8600h
Init: _ccd5_lthree0_core4_thread1_inst6_n169_aliasMSR: 2288_8700h
Init: _ccd5_lthree0_core5_thread0_inst6_n170_aliasMSR: 228A_8600h
Init: _ccd5_lthree0_core5_thread1_inst6_n171_aliasMSR: 228A_8700h
Init: _ccd5_lthree0_core6_thread0_inst6_n172_aliasMSR: 228C_8600h
Init: _ccd5_lthree0_core6_thread1_inst6_n173_aliasMSR: 228C_8700h
Init: _ccd5_lthree0_core7_thread0_inst6_n174_aliasMSR: 228E_8600h
Init: _ccd5_lthree0_core7_thread1_inst6_n175_aliasMSR: 228E_8700h
Init: _ccd5_lthree0_core8_thread0_inst6_n176_aliasMSR: 2290_8600h
Init: _ccd5_lthree0_core8_thread1_inst6_n177_aliasMSR: 2290_8700h
Init: _ccd5_lthree0_core9_thread0_inst6_n178_aliasMSR: 2292_8600h

Init: _ccd5_lthree0_core9_thread1_inst6_n179_aliasMSR: 2292_8700h
Init: _ccd5_lthree0_core10_thread0_inst6_n180_aliasMSR: 2294_8600h
Init: _ccd5_lthree0_core10_thread1_inst6_n181_aliasMSR: 2294_8700h
Init: _ccd5_lthree0_core11_thread0_inst6_n182_aliasMSR: 2296_8600h
Init: _ccd5_lthree0_core11_thread1_inst6_n183_aliasMSR: 2296_8700h
Init: _ccd5_lthree0_core12_thread0_inst6_n184_aliasMSR: 2298_8600h
Init: _ccd5_lthree0_core12_thread1_inst6_n185_aliasMSR: 2298_8700h
Init: _ccd5_lthree0_core13_thread0_inst6_n186_aliasMSR: 229A_8600h
Init: _ccd5_lthree0_core13_thread1_inst6_n187_aliasMSR: 229A_8700h
Init: _ccd5_lthree0_core14_thread0_inst6_n188_aliasMSR: 229C_8600h
Init: _ccd5_lthree0_core14_thread1_inst6_n189_aliasMSR: 229C_8700h
Init: _ccd5_lthree0_core15_thread0_inst6_n190_aliasMSR: 229E_8600h
Init: _ccd5_lthree0_core15_thread1_inst6_n191_aliasMSR: 229E_8700h
Init: _ccd6_lthree0_core0_thread0_inst6_n192_aliasMSR: 2300_8600h
Init: _ccd6_lthree0_core0_thread1_inst6_n193_aliasMSR: 2300_8700h
Init: _ccd6_lthree0_core1_thread0_inst6_n194_aliasMSR: 2302_8600h
Init: _ccd6_lthree0_core1_thread1_inst6_n195_aliasMSR: 2302_8700h
Init: _ccd6_lthree0_core2_thread0_inst6_n196_aliasMSR: 2304_8600h
Init: _ccd6_lthree0_core2_thread1_inst6_n197_aliasMSR: 2304_8700h
Init: _ccd6_lthree0_core3_thread0_inst6_n198_aliasMSR: 2306_8600h
Init: _ccd6_lthree0_core3_thread1_inst6_n199_aliasMSR: 2306_8700h
Init: _ccd6_lthree0_core4_thread0_inst6_n200_aliasMSR: 2308_8600h
Init: _ccd6_lthree0_core4_thread1_inst6_n201_aliasMSR: 2308_8700h
Init: _ccd6_lthree0_core5_thread0_inst6_n202_aliasMSR: 230A_8600h
Init: _ccd6_lthree0_core5_thread1_inst6_n203_aliasMSR: 230A_8700h
Init: _ccd6_lthree0_core6_thread0_inst6_n204_aliasMSR: 230C_8600h
Init: _ccd6_lthree0_core6_thread1_inst6_n205_aliasMSR: 230C_8700h
Init: _ccd6_lthree0_core7_thread0_inst6_n206_aliasMSR: 230E_8600h
Init: _ccd6_lthree0_core7_thread1_inst6_n207_aliasMSR: 230E_8700h
Init: _ccd6_lthree0_core8_thread0_inst6_n208_aliasMSR: 2310_8600h
Init: _ccd6_lthree0_core8_thread1_inst6_n209_aliasMSR: 2310_8700h
Init: _ccd6_lthree0_core9_thread0_inst6_n210_aliasMSR: 2312_8600h
Init: _ccd6_lthree0_core9_thread1_inst6_n211_aliasMSR: 2312_8700h
Init: _ccd6_lthree0_core10_thread0_inst6_n212_aliasMSR: 2314_8600h
Init: _ccd6_lthree0_core10_thread1_inst6_n213_aliasMSR: 2314_8700h
Init: _ccd6_lthree0_core11_thread0_inst6_n214_aliasMSR: 2316_8600h
Init: _ccd6_lthree0_core11_thread1_inst6_n215_aliasMSR: 2316_8700h
Init: _ccd6_lthree0_core12_thread0_inst6_n216_aliasMSR: 2318_8600h
Init: _ccd6_lthree0_core12_thread1_inst6_n217_aliasMSR: 2318_8700h
Init: _ccd6_lthree0_core13_thread0_inst6_n218_aliasMSR: 231A_8600h
Init: _ccd6_lthree0_core13_thread1_inst6_n219_aliasMSR: 231A_8700h
Init: _ccd6_lthree0_core14_thread0_inst6_n220_aliasMSR: 231C_8600h
Init: _ccd6_lthree0_core14_thread1_inst6_n221_aliasMSR: 231C_8700h
Init: _ccd6_lthree0_core15_thread0_inst6_n222_aliasMSR: 231E_8600h
Init: _ccd6_lthree0_core15_thread1_inst6_n223_aliasMSR: 231E_8700h
Init: _ccd7_lthree0_core0_thread0_inst6_n224_aliasMSR: 2380_8600h
Init: _ccd7_lthree0_core0_thread1_inst6_n225_aliasMSR: 2380_8700h
Init: _ccd7_lthree0_core1_thread0_inst6_n226_aliasMSR: 2382_8600h
Init: _ccd7_lthree0_core1_thread1_inst6_n227_aliasMSR: 2382_8700h
Init: _ccd7_lthree0_core2_thread0_inst6_n228_aliasMSR: 2384_8600h
Init: _ccd7_lthree0_core2_thread1_inst6_n229_aliasMSR: 2384_8700h
Init: _ccd7_lthree0_core3_thread0_inst6_n230_aliasMSR: 2386_8600h
Init: _ccd7_lthree0_core3_thread1_inst6_n231_aliasMSR: 2386_8700h
Init: _ccd7_lthree0_core4_thread0_inst6_n232_aliasMSR: 2388_8600h
Init: _ccd7_lthree0_core4_thread1_inst6_n233_aliasMSR: 2388_8700h
Init: _ccd7_lthree0_core5_thread0_inst6_n234_aliasMSR: 238A_8600h
Init: _ccd7_lthree0_core5_thread1_inst6_n235_aliasMSR: 238A_8700h
Init: _ccd7_lthree0_core6_thread0_inst6_n236_aliasMSR: 238C_8600h
Init: _ccd7_lthree0_core6_thread1_inst6_n237_aliasMSR: 238C_8700h
Init: _ccd7_lthree0_core7_thread0_inst6_n238_aliasMSR: 238E_8600h
Init: _ccd7_lthree0_core7_thread1_inst6_n239_aliasMSR: 238E_8700h
Init: _ccd7_lthree0_core8_thread0_inst6_n240_aliasMSR: 2390_8600h
Init: _ccd7_lthree0_core8_thread1_inst6_n241_aliasMSR: 2390_8700h
Init: _ccd7_lthree0_core9_thread0_inst6_n242_aliasMSR: 2392_8600h
Init: _ccd7_lthree0_core9_thread1_inst6_n243_aliasMSR: 2392_8700h

Init: _ccd7_lthree0_core10_thread0_inst6_n244_aliasMSR: 2394_8600h
Init: _ccd7_lthree0_core10_thread1_inst6_n245_aliasMSR: 2394_8700h
Init: _ccd7_lthree0_core11_thread0_inst6_n246_aliasMSR: 2396_8600h
Init: _ccd7_lthree0_core11_thread1_inst6_n247_aliasMSR: 2396_8700h
Init: _ccd7_lthree0_core12_thread0_inst6_n248_aliasMSR: 2398_8600h
Init: _ccd7_lthree0_core12_thread1_inst6_n249_aliasMSR: 2398_8700h
Init: _ccd7_lthree0_core13_thread0_inst6_n250_aliasMSR: 239A_8600h
Init: _ccd7_lthree0_core13_thread1_inst6_n251_aliasMSR: 239A_8700h
Init: _ccd7_lthree0_core14_thread0_inst6_n252_aliasMSR: 239C_8600h
Init: _ccd7_lthree0_core14_thread1_inst6_n253_aliasMSR: 239C_8700h
Init: _ccd7_lthree0_core15_thread0_inst6_n254_aliasMSR: 239E_8600h
Init: _ccd7_lthree0_core15_thread1_inst6_n255_aliasMSR: 239E_8700h
Init: _ccd8_lthree0_core0_thread0_inst6_n256_aliasMSR: 2400_8600h
Init: _ccd8_lthree0_core0_thread1_inst6_n257_aliasMSR: 2400_8700h
Init: _ccd8_lthree0_core1_thread0_inst6_n258_aliasMSR: 2402_8600h
Init: _ccd8_lthree0_core1_thread1_inst6_n259_aliasMSR: 2402_8700h
Init: _ccd8_lthree0_core2_thread0_inst6_n260_aliasMSR: 2404_8600h
Init: _ccd8_lthree0_core2_thread1_inst6_n261_aliasMSR: 2404_8700h
Init: _ccd8_lthree0_core3_thread0_inst6_n262_aliasMSR: 2406_8600h
Init: _ccd8_lthree0_core3_thread1_inst6_n263_aliasMSR: 2406_8700h
Init: _ccd8_lthree0_core4_thread0_inst6_n264_aliasMSR: 2408_8600h
Init: _ccd8_lthree0_core4_thread1_inst6_n265_aliasMSR: 2408_8700h
Init: _ccd8_lthree0_core5_thread0_inst6_n266_aliasMSR: 240A_8600h
Init: _ccd8_lthree0_core5_thread1_inst6_n267_aliasMSR: 240A_8700h
Init: _ccd8_lthree0_core6_thread0_inst6_n268_aliasMSR: 240C_8600h
Init: _ccd8_lthree0_core6_thread1_inst6_n269_aliasMSR: 240C_8700h
Init: _ccd8_lthree0_core7_thread0_inst6_n270_aliasMSR: 240E_8600h
Init: _ccd8_lthree0_core7_thread1_inst6_n271_aliasMSR: 240E_8700h
Init: _ccd8_lthree0_core8_thread0_inst6_n272_aliasMSR: 2410_8600h
Init: _ccd8_lthree0_core8_thread1_inst6_n273_aliasMSR: 2410_8700h
Init: _ccd8_lthree0_core9_thread0_inst6_n274_aliasMSR: 2412_8600h
Init: _ccd8_lthree0_core9_thread1_inst6_n275_aliasMSR: 2412_8700h
Init: _ccd8_lthree0_core10_thread0_inst6_n276_aliasMSR: 2414_8600h
Init: _ccd8_lthree0_core10_thread1_inst6_n277_aliasMSR: 2414_8700h
Init: _ccd8_lthree0_core11_thread0_inst6_n278_aliasMSR: 2416_8600h
Init: _ccd8_lthree0_core11_thread1_inst6_n279_aliasMSR: 2416_8700h
Init: _ccd8_lthree0_core12_thread0_inst6_n280_aliasMSR: 2418_8600h
Init: _ccd8_lthree0_core12_thread1_inst6_n281_aliasMSR: 2418_8700h
Init: _ccd8_lthree0_core13_thread0_inst6_n282_aliasMSR: 241A_8600h
Init: _ccd8_lthree0_core13_thread1_inst6_n283_aliasMSR: 241A_8700h
Init: _ccd8_lthree0_core14_thread0_inst6_n284_aliasMSR: 241C_8600h
Init: _ccd8_lthree0_core14_thread1_inst6_n285_aliasMSR: 241C_8700h
Init: _ccd8_lthree0_core15_thread0_inst6_n286_aliasMSR: 241E_8600h
Init: _ccd8_lthree0_core15_thread1_inst6_n287_aliasMSR: 241E_8700h
Init: _ccd9_lthree0_core0_thread0_inst6_n288_aliasMSR: 2480_8600h
Init: _ccd9_lthree0_core0_thread1_inst6_n289_aliasMSR: 2480_8700h
Init: _ccd9_lthree0_core1_thread0_inst6_n290_aliasMSR: 2482_8600h
Init: _ccd9_lthree0_core1_thread1_inst6_n291_aliasMSR: 2482_8700h
Init: _ccd9_lthree0_core2_thread0_inst6_n292_aliasMSR: 2484_8600h
Init: _ccd9_lthree0_core2_thread1_inst6_n293_aliasMSR: 2484_8700h
Init: _ccd9_lthree0_core3_thread0_inst6_n294_aliasMSR: 2486_8600h
Init: _ccd9_lthree0_core3_thread1_inst6_n295_aliasMSR: 2486_8700h
Init: _ccd9_lthree0_core4_thread0_inst6_n296_aliasMSR: 2488_8600h
Init: _ccd9_lthree0_core4_thread1_inst6_n297_aliasMSR: 2488_8700h
Init: _ccd9_lthree0_core5_thread0_inst6_n298_aliasMSR: 248A_8600h
Init: _ccd9_lthree0_core5_thread1_inst6_n299_aliasMSR: 248A_8700h
Init: _ccd9_lthree0_core6_thread0_inst6_n300_aliasMSR: 248C_8600h
Init: _ccd9_lthree0_core6_thread1_inst6_n301_aliasMSR: 248C_8700h
Init: _ccd9_lthree0_core7_thread0_inst6_n302_aliasMSR: 248E_8600h
Init: _ccd9_lthree0_core7_thread1_inst6_n303_aliasMSR: 248E_8700h
Init: _ccd9_lthree0_core8_thread0_inst6_n304_aliasMSR: 2490_8600h
Init: _ccd9_lthree0_core8_thread1_inst6_n305_aliasMSR: 2490_8700h
Init: _ccd9_lthree0_core9_thread0_inst6_n306_aliasMSR: 2492_8600h
Init: _ccd9_lthree0_core9_thread1_inst6_n307_aliasMSR: 2492_8700h
Init: _ccd9_lthree0_core10_thread0_inst6_n308_aliasMSR: 2494_8600h

Init: _ccd9_lthree0_core10_thread1_inst6_n309_aliasMSR: 2494_8700h
Init: _ccd9_lthree0_core11_thread0_inst6_n310_aliasMSR: 2496_8600h
Init: _ccd9_lthree0_core11_thread1_inst6_n311_aliasMSR: 2496_8700h
Init: _ccd9_lthree0_core12_thread0_inst6_n312_aliasMSR: 2498_8600h
Init: _ccd9_lthree0_core12_thread1_inst6_n313_aliasMSR: 2498_8700h
Init: _ccd9_lthree0_core13_thread0_inst6_n314_aliasMSR: 249A_8600h
Init: _ccd9_lthree0_core13_thread1_inst6_n315_aliasMSR: 249A_8700h
Init: _ccd9_lthree0_core14_thread0_inst6_n316_aliasMSR: 249C_8600h
Init: _ccd9_lthree0_core14_thread1_inst6_n317_aliasMSR: 249C_8700h
Init: _ccd9_lthree0_core15_thread0_inst6_n318_aliasMSR: 249E_8600h
Init: _ccd9_lthree0_core15_thread1_inst6_n319_aliasMSR: 249E_8700h
Init: _ccd10_lthree0_core0_thread0_inst6_n320_aliasMSR: 2500_8600h
Init: _ccd10_lthree0_core0_thread1_inst6_n321_aliasMSR: 2500_8700h
Init: _ccd10_lthree0_core1_thread0_inst6_n322_aliasMSR: 2502_8600h
Init: _ccd10_lthree0_core1_thread1_inst6_n323_aliasMSR: 2502_8700h
Init: _ccd10_lthree0_core2_thread0_inst6_n324_aliasMSR: 2504_8600h
Init: _ccd10_lthree0_core2_thread1_inst6_n325_aliasMSR: 2504_8700h
Init: _ccd10_lthree0_core3_thread0_inst6_n326_aliasMSR: 2506_8600h
Init: _ccd10_lthree0_core3_thread1_inst6_n327_aliasMSR: 2506_8700h
Init: _ccd10_lthree0_core4_thread0_inst6_n328_aliasMSR: 2508_8600h
Init: _ccd10_lthree0_core4_thread1_inst6_n329_aliasMSR: 2508_8700h
Init: _ccd10_lthree0_core5_thread0_inst6_n330_aliasMSR: 250A_8600h
Init: _ccd10_lthree0_core5_thread1_inst6_n331_aliasMSR: 250A_8700h
Init: _ccd10_lthree0_core6_thread0_inst6_n332_aliasMSR: 250C_8600h
Init: _ccd10_lthree0_core6_thread1_inst6_n333_aliasMSR: 250C_8700h
Init: _ccd10_lthree0_core7_thread0_inst6_n334_aliasMSR: 250E_8600h
Init: _ccd10_lthree0_core7_thread1_inst6_n335_aliasMSR: 250E_8700h
Init: _ccd10_lthree0_core8_thread0_inst6_n336_aliasMSR: 2510_8600h
Init: _ccd10_lthree0_core8_thread1_inst6_n337_aliasMSR: 2510_8700h
Init: _ccd10_lthree0_core9_thread0_inst6_n338_aliasMSR: 2512_8600h
Init: _ccd10_lthree0_core9_thread1_inst6_n339_aliasMSR: 2512_8700h
Init: _ccd10_lthree0_core10_thread0_inst6_n340_aliasMSR: 2514_8600h
Init: _ccd10_lthree0_core10_thread1_inst6_n341_aliasMSR: 2514_8700h
Init: _ccd10_lthree0_core11_thread0_inst6_n342_aliasMSR: 2516_8600h
Init: _ccd10_lthree0_core11_thread1_inst6_n343_aliasMSR: 2516_8700h
Init: _ccd10_lthree0_core12_thread0_inst6_n344_aliasMSR: 2518_8600h
Init: _ccd10_lthree0_core12_thread1_inst6_n345_aliasMSR: 2518_8700h
Init: _ccd10_lthree0_core13_thread0_inst6_n346_aliasMSR: 251A_8600h
Init: _ccd10_lthree0_core13_thread1_inst6_n347_aliasMSR: 251A_8700h
Init: _ccd10_lthree0_core14_thread0_inst6_n348_aliasMSR: 251C_8600h
Init: _ccd10_lthree0_core14_thread1_inst6_n349_aliasMSR: 251C_8700h
Init: _ccd10_lthree0_core15_thread0_inst6_n350_aliasMSR: 251E_8600h
Init: _ccd10_lthree0_core15_thread1_inst6_n351_aliasMSR: 251E_8700h
Init: _ccd11_lthree0_core0_thread0_inst6_n352_aliasMSR: 2580_8600h
Init: _ccd11_lthree0_core0_thread1_inst6_n353_aliasMSR: 2580_8700h
Init: _ccd11_lthree0_core1_thread0_inst6_n354_aliasMSR: 2582_8600h
Init: _ccd11_lthree0_core1_thread1_inst6_n355_aliasMSR: 2582_8700h
Init: _ccd11_lthree0_core2_thread0_inst6_n356_aliasMSR: 2584_8600h
Init: _ccd11_lthree0_core2_thread1_inst6_n357_aliasMSR: 2584_8700h
Init: _ccd11_lthree0_core3_thread0_inst6_n358_aliasMSR: 2586_8600h
Init: _ccd11_lthree0_core3_thread1_inst6_n359_aliasMSR: 2586_8700h
Init: _ccd11_lthree0_core4_thread0_inst6_n360_aliasMSR: 2588_8600h
Init: _ccd11_lthree0_core4_thread1_inst6_n361_aliasMSR: 2588_8700h
Init: _ccd11_lthree0_core5_thread0_inst6_n362_aliasMSR: 258A_8600h
Init: _ccd11_lthree0_core5_thread1_inst6_n363_aliasMSR: 258A_8700h
Init: _ccd11_lthree0_core6_thread0_inst6_n364_aliasMSR: 258C_8600h
Init: _ccd11_lthree0_core6_thread1_inst6_n365_aliasMSR: 258C_8700h
Init: _ccd11_lthree0_core7_thread0_inst6_n366_aliasMSR: 258E_8600h
Init: _ccd11_lthree0_core7_thread1_inst6_n367_aliasMSR: 258E_8700h
Init: _ccd11_lthree0_core8_thread0_inst6_n368_aliasMSR: 2590_8600h
Init: _ccd11_lthree0_core8_thread1_inst6_n369_aliasMSR: 2590_8700h
Init: _ccd11_lthree0_core9_thread0_inst6_n370_aliasMSR: 2592_8600h
Init: _ccd11_lthree0_core9_thread1_inst6_n371_aliasMSR: 2592_8700h
Init: _ccd11_lthree0_core10_thread0_inst6_n372_aliasMSR: 2594_8600h
Init: _ccd11_lthree0_core10_thread1_inst6_n373_aliasMSR: 2594_8700h

Init: _ccd11_lthree0_core11_thread0_inst6_n374_aliasMSR: 2596_8600h
Init: _ccd11_lthree0_core11_thread1_inst6_n375_aliasMSR: 2596_8700h
Init: _ccd11_lthree0_core12_thread0_inst6_n376_aliasMSR: 2598_8600h
Init: _ccd11_lthree0_core12_thread1_inst6_n377_aliasMSR: 2598_8700h
Init: _ccd11_lthree0_core13_thread0_inst6_n378_aliasMSR: 259A_8600h
Init: _ccd11_lthree0_core13_thread1_inst6_n379_aliasMSR: 259A_8700h
Init: _ccd11_lthree0_core14_thread0_inst6_n380_aliasMSR: 259C_8600h
Init: _ccd11_lthree0_core14_thread1_inst6_n381_aliasMSR: 259C_8700h
Init: _ccd11_lthree0_core15_thread0_inst6_n382_aliasMSR: 259E_8600h
Init: _ccd11_lthree0_core15_thread1_inst6_n383_aliasMSR: 259E_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](#)

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst6_n[383:0]_aliasMSR; MSRC000_2066

Bits	Description
63:27	Reserved.
26:24	ErrorPriority. 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	Length. 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	ErrorInformation. 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_FP].

Table 69: 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.

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst6_n[383:0]_aliasMSR; MSRC001_0406

Bits	Description
63:8	Reserved.
7	KRF. Read-write. Reset: 0. Physical K mask register file (KRF) parity error.
6	HWA. Read-write. Reset: 0. Hardware assertion.
5	SRF. Read-write. Reset: 0. Status register file (SRF) parity error.
4	RQ. Read-write. Reset: 0. Retire queue (RQ) parity error.
3	NSQ. Read-write. Reset: 0. NSQ parity error.
2	SCH. Read-write. Reset: 0. Schedule queue parity error.
1	FL. Read-write. Reset: 0. Freelist (FL) parity error.
0	PRF. 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](#)

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst6_n[383:0]_aliasMSR; MSRC000_206E

Bits	Description
63:0	Syndrome. 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](#)

_ccd[11:0]_lthree0_core[15:0]_thread[1:0]_inst6_n[383:0]_aliasMSR; MSRC000_206F

Bits	Description
63:0	Syndrome. 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::Msrr::MCG_CTL. Does not affect error detection, correction, or logging.	
_ccd[11:0]_lthree0_inst[15,7]_n[184,176,168,160,152,144,136,128,120,112,104,96,88,80,72,64,56,48,40,32,24,16,8,0]	aliasMSRLEGACY; MSR0000_041C
_ccd[11:0]_lthree0_inst[16,8]_n[185,177,169,161,153,145,137,129,121,113,105,97,89,81,73,65,57,49,41,33,25,17,9,1]	aliasMSRLEGACY; MSR0000_0420
_ccd[11:0]_lthree0_inst[17,9]_n[186,178,170,162,154,146,138,130,122,114,106,98,90,82,74,66,58,50,42,34,26,18,10,2]	aliasMSRLEGACY; MSR0000_0424
_ccd[11:0]_lthree0_inst[18,10]_n[187,179,171,163,155,147,139,131,123,115,107,99,91,83,75,67,59,51,43,35,27,19,11,3]	aliasMSRLEGACY; MSR0000_0428
_ccd[11:0]_lthree0_inst[19,11]_n[188,180,172,164,156,148,140,132,124,116,108,100,92,84,76,68,60,52,44,36,28,20,12,4]	aliasMSRLEGACY; MSR0000_042C
_ccd[11:0]_lthree0_inst[20,12]_n[189,181,173,165,157,149,141,133,125,117,109,101,93,85,77,69,61,53,45,37,29,21,13,5]	aliasMSRLEGACY; MSR0000_0430
_ccd[11:0]_lthree0_inst[21,13]_n[190,182,174,166,158,150,142,134,126,118,110,102,94,86,78,70,62,54,46,38,30,22,14,6]	aliasMSRLEGACY; MSR0000_0434
_ccd[11:0]_lthree0_inst[22,14]_n[191,183,175,167,159,151,143,135,127,119,111,103,95,87,79,71,63,55,47,39,31,23,15,7]	aliasMSRLEGACY; MSR0000_0438
_ccd[11:0]_lthree0_inst[15,7]_n[184,176,168,160,152,144,136,128,120,112,104,96,88,80,72,64,56,48,40,32,24,16,8,0]	aliasMSR; MSRC000_2070
_ccd[11:0]_lthree0_inst[16,8]_n[185,177,169,161,153,145,137,129,121,113,105,97,89,81,73,65,57,49,41,33,25,17,9,1]	aliasMSR; MSRC000_2080
_ccd[11:0]_lthree0_inst[17,9]_n[186,178,170,162,154,146,138,130,122,114,106,98,90,82,74,66,58,50,42,34,26,18,10,2]	aliasMSR; MSRC000_2090
_ccd[11:0]_lthree0_inst[18,10]_n[187,179,171,163,155,147,139,131,123,115,107,99,91,83,75,67,59,51,43,35,27,19,11,3]	aliasMSR; MSRC000_20A0
_ccd[11:0]_lthree0_inst[19,11]_n[188,180,172,164,156,148,140,132,124,116,108,100,92,84,76,68,60,52,44,36,28,20,12,4]	aliasMSR; MSRC000_20B0
_ccd[11:0]_lthree0_inst[20,12]_n[189,181,173,165,157,149,141,133,125,117,109,101,93,85,77,69,61,53,45,37,29,21,13,5]	aliasMSR; MSRC000_20C0
_ccd[11:0]_lthree0_inst[21,13]_n[190,182,174,166,158,150,142,134,126,118,110,102,94,86,78,70,62,54,46,38,30,22,14,6]	aliasMSR; MSRC000_20D0
_ccd[11:0]_lthree0_inst[22,14]_n[191,183,175,167,159,151,143,135,127,119,111,103,95,87,79,71,63,55,47,39,31,23,15,7]	aliasMSR; MSRC000_20E0
Bits	Description
63:10	Reserved.
9	DsmMce. Read-write. Reset: 0. Machine check error initiated by DSM action
8	XiWcbParityPoison. Read-write. Reset: 0. Xi Wcb Parity Poison Creation Event
7	Hwa. Read-write. Reset: 0. L3 Hardware Assertion.
6	XiVictimQueue. Read-write. Reset: 0. L3 Victim Queue Data Fabric Error.
5	SdpParity. Read-write. Reset: 0. SDP Parity Error from XI.
4	DataArray. Read-write. Reset: 0. L3M Data ECC Error.
3	MultiHitTag. Read-write. Reset: 0. L3M Tag Multi-way-hit Error.
2	Tag. Read-write. Reset: 0. L3M Tag ECC Error.
1	MultiHitShadowTag. Read-write. Reset: 0. Shadow Tag Macro Multi-way-hit Error.
0	ShadowTag. 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.

_ccd[11:0]_lthree0_inst[15,7]_n[184,176,168,160,152,144,136,128,120,112,104,96,88,80,72,64,56,48,40,32,24,16,8,0]	aliasMSRLEGACY; MSR0000_041D
_ccd[11:0]_lthree0_inst[16,8]_n[185,177,169,161,153,145,137,129,121,113,105,97,89,81,73,65,57,49,41,33,25,17,9,1]	aliasMSRLEGACY; MSR0000_0421
_ccd[11:0]_lthree0_inst[17,9]_n[186,178,170,162,154,146,138,130,122,114,106,98,90,82,74,66,58,50,42,34,26,18,10,2]	aliasMSRLEGACY; MSR0000_0425
_ccd[11:0]_lthree0_inst[18,10]_n[187,179,171,163,155,147,139,131,123,115,107,99,91,83,75,67,59,51,43,35,27,19,11,3]	aliasMSRLEGACY; MSR0000_0429
_ccd[11:0]_lthree0_inst[19,11]_n[188,180,172,164,156,148,140,132,124,116,108,100,92,84,76,68,60,52,44,36,28,20,12,4]	aliasMSRLEGACY; MSR0000_042D
_ccd[11:0]_lthree0_inst[20,12]_n[189,181,173,165,157,149,141,133,125,117,109,101,93,85,77,69,61,53,45,37,29,21,13,5]	aliasMSRLEGACY; MSR0000_0431
_ccd[11:0]_lthree0_inst[21,13]_n[190,182,174,166,158,150,142,134,126,118,110,102,94,86,78,70,62,54,46,38,30,22,14,6]	aliasMSRLEGACY; MSR0000_0435
_ccd[11:0]_lthree0_inst[22,14]_n[191,183,175,167,159,151,143,135,127,119,111,103,95,87,79,71,63,55,47,39,31,23,15,7]	aliasMSRLEGACY; MSR0000_0439
_ccd[11:0]_lthree0_inst[15,7]_n[184,176,168,160,152,144,136,128,120,112,104,96,88,80,72,64,56,48,40,32,24,16,8,0]	aliasMSR; MSRC000_2071
_ccd[11:0]_lthree0_inst[16,8]_n[185,177,169,161,153,145,137,129,121,113,105,97,89,81,73,65,57,49,41,33,25,17,9,1]	aliasMSR; MSRC000_2081
_ccd[11:0]_lthree0_inst[17,9]_n[186,178,170,162,154,146,138,130,122,114,106,98,90,82,74,66,58,50,42,34,26,18,10,2]	aliasMSR; MSRC000_2091
_ccd[11:0]_lthree0_inst[18,10]_n[187,179,171,163,155,147,139,131,123,115,107,99,91,83,75,67,59,51,43,35,27,19,11,3]	aliasMSR; MSRC000_20A1
_ccd[11:0]_lthree0_inst[19,11]_n[188,180,172,164,156,148,140,132,124,116,108,100,92,84,76,68,60,52,44,36,28,20,12,4]	aliasMSR; MSRC000_20B1
_ccd[11:0]_lthree0_inst[20,12]_n[189,181,173,165,157,149,141,133,125,117,109,101,93,85,77,69,61,53,45,37,29,21,13,5]	aliasMSR; MSRC000_20C1
_ccd[11:0]_lthree0_inst[21,13]_n[190,182,174,166,158,150,142,134,126,118,110,102,94,86,78,70,62,54,46,38,30,22,14,6]	aliasMSR; MSRC000_20D1
_ccd[11:0]_lthree0_inst[22,14]_n[191,183,175,167,159,151,143,135,127,119,111,103,95,87,79,71,63,55,47,39,31,23,15,7]	aliasMSR; MSRC000_20E1

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. 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	MiscV. 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	AddrV. 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	PCC. 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	ErrCoreIdVal. 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	TCC. 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	RESERV54. 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	SyndV. 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	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. 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	RESERV22. 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	ErrorCodeExt. 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	ErrorCode. 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 70: 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.

_ccd[11:0]_lthree0_inst[15,7]_n[184,176,168,160,152,144,136,128,120,112,104,96,88,80,72,64,56,48,40,32,24,16,8,0]_aliasMSRLEGACY; MSR0000_041E
_ccd[11:0]_lthree0_inst[16,8]_n[185,177,169,161,153,145,137,129,121,113,105,97,89,81,73,65,57,49,41,33,25,17,9,1]_aliasMSRLEGACY; MSR0000_0422
_ccd[11:0]_lthree0_inst[17,9]_n[186,178,170,162,154,146,138,130,122,114,106,98,90,82,74,66,58,50,42,34,26,18,10,2]_aliasMSRLEGACY; MSR0000_0426
_ccd[11:0]_lthree0_inst[18,10]_n[187,179,171,163,155,147,139,131,123,115,107,99,91,83,75,67,59,51,43,35,27,19,11,3]_aliasMSRLEGACY; MSR0000_042A
_ccd[11:0]_lthree0_inst[19,11]_n[188,180,172,164,156,148,140,132,124,116,108,100,92,84,76,68,60,52,44,36,28,20,12,4]_aliasMSRLEGACY; MSR0000_042E
_ccd[11:0]_lthree0_inst[20,12]_n[189,181,173,165,157,149,141,133,125,117,109,101,93,85,77,69,61,53,45,37,29,21,13,5]_aliasMSRLEGACY; MSR0000_0432
_ccd[11:0]_lthree0_inst[21,13]_n[190,182,174,166,158,150,142,134,126,118,110,102,94,86,78,70,62,54,46,38,30,22,14,6]_aliasMSRLEGACY; MSR0000_0436
_ccd[11:0]_lthree0_inst[22,14]_n[191,183,175,167,159,151,143,135,127,119,111,103,95,87,79,71,63,55,47,39,31,23,15,7]_aliasMSRLEGACY; MSR0000_043A
_ccd[11:0]_lthree0_inst[15,7]_n[184,176,168,160,152,144,136,128,120,112,104,96,88,80,72,64,56,48,40,32,24,16,8,0]_aliasMSR; MSRC000_2072
_ccd[11:0]_lthree0_inst[16,8]_n[185,177,169,161,153,145,137,129,121,113,105,97,89,81,73,65,57,49,41,33,25,17,9,1]_aliasMSR; MSRC000_2082
_ccd[11:0]_lthree0_inst[17,9]_n[186,178,170,162,154,146,138,130,122,114,106,98,90,82,74,66,58,50,42,34,26,18,10,2]_aliasMSR; MSRC000_2092
_ccd[11:0]_lthree0_inst[18,10]_n[187,179,171,163,155,147,139,131,123,115,107,99,91,83,75,67,59,51,43,35,27,19,11,3]_aliasMSR; MSRC000_20A2
_ccd[11:0]_lthree0_inst[19,11]_n[188,180,172,164,156,148,140,132,124,116,108,100,92,84,76,68,60,52,44,36,28,20,12,4]_aliasMSR; MSRC000_20B2
_ccd[11:0]_lthree0_inst[20,12]_n[189,181,173,165,157,149,141,133,125,117,109,101,93,85,77,69,61,53,45,37,29,21,13,5]_aliasMSR; MSRC000_20C2
_ccd[11:0]_lthree0_inst[21,13]_n[190,182,174,166,158,150,142,134,126,118,110,102,94,86,78,70,62,54,46,38,30,22,14,6]_aliasMSR; MSRC000_20D2
_ccd[11:0]_lthree0_inst[22,14]_n[191,183,175,167,159,151,143,135,127,119,111,103,95,87,79,71,63,55,47,39,31,23,15,7]_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 71: 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	[64:0]	Reserved

MSR0000_041F...MSRC000_20E3 [L3 Machine Check Miscellaneous 0] (MCA::L3::MCA_MISC0_L3)

Log miscellaneous information associated with errors.

_ccd[11:0]_lthree0_inst[15,7]_n[184,176,168,160,152,144,136,128,120,112,104,96,88,80,72,64,56,48,40,32,24,16,8,0]_aliasMSRLEGACY; MSR0000_041F	
_ccd[11:0]_lthree0_inst[16,8]_n[185,177,169,161,153,145,137,129,121,113,105,97,89,81,73,65,57,49,41,33,25,17,9,1]_aliasMSRLEGACY; MSR0000_0423	
_ccd[11:0]_lthree0_inst[17,9]_n[186,178,170,162,154,146,138,130,122,114,106,98,90,82,74,66,58,50,42,34,26,18,10,2]_aliasMSRLEGACY; MSR0000_0427	
_ccd[11:0]_lthree0_inst[18,10]_n[187,179,171,163,155,147,139,131,123,115,107,99,91,83,75,67,59,51,43,35,27,19,11,3]_aliasMSRLEGACY; MSR0000_042B	
_ccd[11:0]_lthree0_inst[19,11]_n[188,180,172,164,156,148,140,132,124,116,108,100,92,84,76,68,60,52,44,36,28,20,12,4]_aliasMSRLEGACY; MSR0000_042F	
_ccd[11:0]_lthree0_inst[20,12]_n[189,181,173,165,157,149,141,133,125,117,109,101,93,85,77,69,61,53,45,37,29,21,13,5]_aliasMSRLEGACY; MSR0000_0433	
_ccd[11:0]_lthree0_inst[21,13]_n[190,182,174,166,158,150,142,134,126,118,110,102,94,86,78,70,62,54,46,38,30,22,14,6]_aliasMSRLEGACY; MSR0000_0437	
_ccd[11:0]_lthree0_inst[22,14]_n[191,183,175,167,159,151,143,135,127,119,111,103,95,87,79,71,63,55,47,39,31,23,15,7]_aliasMSRLEGACY; MSR0000_043B	
_ccd[11:0]_lthree0_inst[15,7]_n[184,176,168,160,152,144,136,128,120,112,104,96,88,80,72,64,56,48,40,32,24,16,8,0]_aliasMSR; MSRC000_2073	
_ccd[11:0]_lthree0_inst[16,8]_n[185,177,169,161,153,145,137,129,121,113,105,97,89,81,73,65,57,49,41,33,25,17,9,1]_aliasMSR; MSRC000_2083	
_ccd[11:0]_lthree0_inst[17,9]_n[186,178,170,162,154,146,138,130,122,114,106,98,90,82,74,66,58,50,42,34,26,18,10,2]_aliasMSR; MSRC000_2093	
_ccd[11:0]_lthree0_inst[18,10]_n[187,179,171,163,155,147,139,131,123,115,107,99,91,83,75,67,59,51,43,35,27,19,11,3]_aliasMSR; MSRC000_20A3	
_ccd[11:0]_lthree0_inst[19,11]_n[188,180,172,164,156,148,140,132,124,116,108,100,92,84,76,68,60,52,44,36,28,20,12,4]_aliasMSR; MSRC000_20B3	
_ccd[11:0]_lthree0_inst[20,12]_n[189,181,173,165,157,149,141,133,125,117,109,101,93,85,77,69,61,53,45,37,29,21,13,5]_aliasMSR; MSRC000_20C3	
_ccd[11:0]_lthree0_inst[21,13]_n[190,182,174,166,158,150,142,134,126,118,110,102,94,86,78,70,62,54,46,38,30,22,14,6]_aliasMSR; MSRC000_20D3	
_ccd[11:0]_lthree0_inst[22,14]_n[191,183,175,167,159,151,143,135,127,119,111,103,95,87,79,71,63,55,47,39,31,23,15,7]_aliasMSR; MSRC000_20E3	
Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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	LvtOffset. 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	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn] !MCA::L3::MCA_MISC0_L3[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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	Ovrflw. 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	ErrCnt. 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	BlkPtr. 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.

_ccd[11:0]_lthree0_inst[15,7]_n[184,176,168,160,152,144,136,128,120,112,104,96,88,80,72,64,56,48,40,32,24,16,8,0]_aliasMSR; MSRC000_2074

_ccd[11:0]_lthree0_inst[16,8]_n[185,177,169,161,153,145,137,129,121,113,105,97,89,81,73,65,57,49,41,33,25,17,9,1]_aliasMSR; MSRC000_2084

_ccd[11:0]_lthree0_inst[17,9]_n[186,178,170,162,154,146,138,130,122,114,106,98,90,82,74,66,58,50,42,34,26,18,10,2]_aliasMSR; MSRC000_2094

_ccd[11:0]_lthree0_inst[18,10]_n[187,179,171,163,155,147,139,131,123,115,107,99,91,83,75,67,59,51,43,35,27,19,11,3]_aliasMSR; MSRC000_20A4

_ccd[11:0]_lthree0_inst[19,11]_n[188,180,172,164,156,148,140,132,124,116,108,100,92,84,76,68,60,52,44,36,28,20,12,4]_aliasMSR; MSRC000_20B4

_ccd[11:0]_lthree0_inst[20,12]_n[189,181,173,165,157,149,141,133,125,117,109,101,93,85,77,69,61,53,45,37,29,21,13,5]_aliasMSR; MSRC000_20C4

_ccd[11:0]_lthree0_inst[21,13]_n[190,182,174,166,158,150,142,134,126,118,110,102,94,86,78,70,62,54,46,38,30,22,14,6]_aliasMSR; MSRC000_20D4

_ccd[11:0]_lthree0_inst[22,14]_n[191,183,175,167,159,151,143,135,127,119,111,103,95,87,79,71,63,55,47,39,31,23,15,7]_aliasMSR; MSRC000_20E4

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	LogDeferredInMcaStat. 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	McaXEnable. 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	IntPresent. 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	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::L3::MCA_CONFIG_L3[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. 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	DeferredErrorLoggingSupported. 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	McaX. 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.

_ccd[11:0]_lthree0_inst[15,7]_n[184,176,168,160,152,144,136,128,120,112,104,96,88,80,72,64,56,48,40,32,24,16,8,0]_aliasMSR; MSRC000_2075

_ccd[11:0]_lthree0_inst[16,8]_n[185,177,169,161,153,145,137,129,121,113,105,97,89,81,73,65,57,49,41,33,25,17,9,1]_aliasMSR; MSRC000_2085

_ccd[11:0]_lthree0_inst[17,9]_n[186,178,170,162,154,146,138,130,122,114,106,98,90,82,74,66,58,50,42,34,26,18,10,2]_aliasMSR; MSRC000_2095

_ccd[11:0]_lthree0_inst[18,10]_n[187,179,171,163,155,147,139,131,123,115,107,99,91,83,75,67,59,51,43,35,27,19,11,3]_aliasMSR; MSRC000_20A5

_ccd[11:0]_lthree0_inst[19,11]_n[188,180,172,164,156,148,140,132,124,116,108,100,92,84,76,68,60,52,44,36,28,20,12,4]_aliasMSR; MSRC000_20B5

_ccd[11:0]_lthree0_inst[20,12]_n[189,181,173,165,157,149,141,133,125,117,109,101,93,85,77,69,61,53,45,37,29,21,13,5]_aliasMSR; MSRC000_20C5

_ccd[11:0]_lthree0_inst[21,13]_n[190,182,174,166,158,150,142,134,126,118,110,102,94,86,78,70,62,54,46,38,30,22,14,6]_aliasMSR; MSRC000_20D5

_ccd[11:0]_lthree0_inst[22,14]_n[191,183,175,167,159,151,143,135,127,119,111,103,95,87,79,71,63,55,47,39,31,23,15,7]_aliasMSR; MSRC000_20E5

Bits	Description
63:48	McaType. Read-only. Reset: 0007h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi. 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	HardwareID. Read-only. Reset: 0B0h. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId. Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register. Init: _ccd0_lthree0_inst7_n0_aliasMSR: 2034_0000h Init: _ccd0_lthree0_inst8_n1_aliasMSR: 2034_1000h Init: _ccd0_lthree0_inst9_n2_aliasMSR: 2034_2000h Init: _ccd0_lthree0_inst10_n3_aliasMSR: 2034_3000h Init: _ccd0_lthree0_inst11_n4_aliasMSR: 2034_4000h Init: _ccd0_lthree0_inst12_n5_aliasMSR: 2034_5000h Init: _ccd0_lthree0_inst13_n6_aliasMSR: 2034_6000h Init: _ccd0_lthree0_inst14_n7_aliasMSR: 2034_7000h Init: _ccd0_lthree0_inst15_n8_aliasMSR: 2034_8000h Init: _ccd0_lthree0_inst16_n9_aliasMSR: 2034_9000h Init: _ccd0_lthree0_inst17_n10_aliasMSR: 2034_A000h Init: _ccd0_lthree0_inst18_n11_aliasMSR: 2034_B000h Init: _ccd0_lthree0_inst19_n12_aliasMSR: 2034_C000h Init: _ccd0_lthree0_inst20_n13_aliasMSR: 2034_D000h Init: _ccd0_lthree0_inst21_n14_aliasMSR: 2034_E000h Init: _ccd0_lthree0_inst22_n15_aliasMSR: 2034_F000h Init: _ccd1_lthree0_inst7_n16_aliasMSR: 20B4_0000h Init: _ccd1_lthree0_inst8_n17_aliasMSR: 20B4_1000h Init: _ccd1_lthree0_inst9_n18_aliasMSR: 20B4_2000h Init: _ccd1_lthree0_inst10_n19_aliasMSR: 20B4_3000h Init: _ccd1_lthree0_inst11_n20_aliasMSR: 20B4_4000h Init: _ccd1_lthree0_inst12_n21_aliasMSR: 20B4_5000h Init: _ccd1_lthree0_inst13_n22_aliasMSR: 20B4_6000h Init: _ccd1_lthree0_inst14_n23_aliasMSR: 20B4_7000h Init: _ccd1_lthree0_inst15_n24_aliasMSR: 20B4_8000h Init: _ccd1_lthree0_inst16_n25_aliasMSR: 20B4_9000h Init: _ccd1_lthree0_inst17_n26_aliasMSR: 20B4_A000h Init: _ccd1_lthree0_inst18_n27_aliasMSR: 20B4_B000h Init: _ccd1_lthree0_inst19_n28_aliasMSR: 20B4_C000h Init: _ccd1_lthree0_inst20_n29_aliasMSR: 20B4_D000h Init: _ccd1_lthree0_inst21_n30_aliasMSR: 20B4_E000h Init: _ccd1_lthree0_inst22_n31_aliasMSR: 20B4_F000h Init: _ccd2_lthree0_inst7_n32_aliasMSR: 2134_0000h Init: _ccd2_lthree0_inst8_n33_aliasMSR: 2134_1000h Init: _ccd2_lthree0_inst9_n34_aliasMSR: 2134_2000h Init: _ccd2_lthree0_inst10_n35_aliasMSR: 2134_3000h Init: _ccd2_lthree0_inst11_n36_aliasMSR: 2134_4000h Init: _ccd2_lthree0_inst12_n37_aliasMSR: 2134_5000h Init: _ccd2_lthree0_inst13_n38_aliasMSR: 2134_6000h Init: _ccd2_lthree0_inst14_n39_aliasMSR: 2134_7000h Init: _ccd2_lthree0_inst15_n40_aliasMSR: 2134_8000h Init: _ccd2_lthree0_inst16_n41_aliasMSR: 2134_9000h

Init: _ccd2_lthree0_inst17_n42_aliasMSR: 2134_A000h
Init: _ccd2_lthree0_inst18_n43_aliasMSR: 2134_B000h
Init: _ccd2_lthree0_inst19_n44_aliasMSR: 2134_C000h
Init: _ccd2_lthree0_inst20_n45_aliasMSR: 2134_D000h
Init: _ccd2_lthree0_inst21_n46_aliasMSR: 2134_E000h
Init: _ccd2_lthree0_inst22_n47_aliasMSR: 2134_F000h
Init: _ccd3_lthree0_inst7_n48_aliasMSR: 21B4_0000h
Init: _ccd3_lthree0_inst8_n49_aliasMSR: 21B4_1000h
Init: _ccd3_lthree0_inst9_n50_aliasMSR: 21B4_2000h
Init: _ccd3_lthree0_inst10_n51_aliasMSR: 21B4_3000h
Init: _ccd3_lthree0_inst11_n52_aliasMSR: 21B4_4000h
Init: _ccd3_lthree0_inst12_n53_aliasMSR: 21B4_5000h
Init: _ccd3_lthree0_inst13_n54_aliasMSR: 21B4_6000h
Init: _ccd3_lthree0_inst14_n55_aliasMSR: 21B4_7000h
Init: _ccd3_lthree0_inst15_n56_aliasMSR: 21B4_8000h
Init: _ccd3_lthree0_inst16_n57_aliasMSR: 21B4_9000h
Init: _ccd3_lthree0_inst17_n58_aliasMSR: 21B4_A000h
Init: _ccd3_lthree0_inst18_n59_aliasMSR: 21B4_B000h
Init: _ccd3_lthree0_inst19_n60_aliasMSR: 21B4_C000h
Init: _ccd3_lthree0_inst20_n61_aliasMSR: 21B4_D000h
Init: _ccd3_lthree0_inst21_n62_aliasMSR: 21B4_E000h
Init: _ccd3_lthree0_inst22_n63_aliasMSR: 21B4_F000h
Init: _ccd4_lthree0_inst7_n64_aliasMSR: 2234_0000h
Init: _ccd4_lthree0_inst8_n65_aliasMSR: 2234_1000h
Init: _ccd4_lthree0_inst9_n66_aliasMSR: 2234_2000h
Init: _ccd4_lthree0_inst10_n67_aliasMSR: 2234_3000h
Init: _ccd4_lthree0_inst11_n68_aliasMSR: 2234_4000h
Init: _ccd4_lthree0_inst12_n69_aliasMSR: 2234_5000h
Init: _ccd4_lthree0_inst13_n70_aliasMSR: 2234_6000h
Init: _ccd4_lthree0_inst14_n71_aliasMSR: 2234_7000h
Init: _ccd4_lthree0_inst15_n72_aliasMSR: 2234_8000h
Init: _ccd4_lthree0_inst16_n73_aliasMSR: 2234_9000h
Init: _ccd4_lthree0_inst17_n74_aliasMSR: 2234_A000h
Init: _ccd4_lthree0_inst18_n75_aliasMSR: 2234_B000h
Init: _ccd4_lthree0_inst19_n76_aliasMSR: 2234_C000h
Init: _ccd4_lthree0_inst20_n77_aliasMSR: 2234_D000h
Init: _ccd4_lthree0_inst21_n78_aliasMSR: 2234_E000h
Init: _ccd4_lthree0_inst22_n79_aliasMSR: 2234_F000h
Init: _ccd5_lthree0_inst7_n80_aliasMSR: 22B4_0000h
Init: _ccd5_lthree0_inst8_n81_aliasMSR: 22B4_1000h
Init: _ccd5_lthree0_inst9_n82_aliasMSR: 22B4_2000h
Init: _ccd5_lthree0_inst10_n83_aliasMSR: 22B4_3000h
Init: _ccd5_lthree0_inst11_n84_aliasMSR: 22B4_4000h
Init: _ccd5_lthree0_inst12_n85_aliasMSR: 22B4_5000h
Init: _ccd5_lthree0_inst13_n86_aliasMSR: 22B4_6000h
Init: _ccd5_lthree0_inst14_n87_aliasMSR: 22B4_7000h
Init: _ccd5_lthree0_inst15_n88_aliasMSR: 22B4_8000h
Init: _ccd5_lthree0_inst16_n89_aliasMSR: 22B4_9000h
Init: _ccd5_lthree0_inst17_n90_aliasMSR: 22B4_A000h
Init: _ccd5_lthree0_inst18_n91_aliasMSR: 22B4_B000h
Init: _ccd5_lthree0_inst19_n92_aliasMSR: 22B4_C000h
Init: _ccd5_lthree0_inst20_n93_aliasMSR: 22B4_D000h
Init: _ccd5_lthree0_inst21_n94_aliasMSR: 22B4_E000h
Init: _ccd5_lthree0_inst22_n95_aliasMSR: 22B4_F000h
Init: _ccd6_lthree0_inst7_n96_aliasMSR: 2334_0000h
Init: _ccd6_lthree0_inst8_n97_aliasMSR: 2334_1000h
Init: _ccd6_lthree0_inst9_n98_aliasMSR: 2334_2000h
Init: _ccd6_lthree0_inst10_n99_aliasMSR: 2334_3000h
Init: _ccd6_lthree0_inst11_n100_aliasMSR: 2334_4000h
Init: _ccd6_lthree0_inst12_n101_aliasMSR: 2334_5000h
Init: _ccd6_lthree0_inst13_n102_aliasMSR: 2334_6000h
Init: _ccd6_lthree0_inst14_n103_aliasMSR: 2334_7000h
Init: _ccd6_lthree0_inst15_n104_aliasMSR: 2334_8000h
Init: _ccd6_lthree0_inst16_n105_aliasMSR: 2334_9000h
Init: _ccd6_lthree0_inst17_n106_aliasMSR: 2334_A000h

Init: _ccd6_lthree0_inst18_n107_aliasMSR: 2334_B000h
Init: _ccd6_lthree0_inst19_n108_aliasMSR: 2334_C000h
Init: _ccd6_lthree0_inst20_n109_aliasMSR: 2334_D000h
Init: _ccd6_lthree0_inst21_n110_aliasMSR: 2334_E000h
Init: _ccd6_lthree0_inst22_n111_aliasMSR: 2334_F000h
Init: _ccd7_lthree0_inst7_n112_aliasMSR: 23B4_0000h
Init: _ccd7_lthree0_inst8_n113_aliasMSR: 23B4_1000h
Init: _ccd7_lthree0_inst9_n114_aliasMSR: 23B4_2000h
Init: _ccd7_lthree0_inst10_n115_aliasMSR: 23B4_3000h
Init: _ccd7_lthree0_inst11_n116_aliasMSR: 23B4_4000h
Init: _ccd7_lthree0_inst12_n117_aliasMSR: 23B4_5000h
Init: _ccd7_lthree0_inst13_n118_aliasMSR: 23B4_6000h
Init: _ccd7_lthree0_inst14_n119_aliasMSR: 23B4_7000h
Init: _ccd7_lthree0_inst15_n120_aliasMSR: 23B4_8000h
Init: _ccd7_lthree0_inst16_n121_aliasMSR: 23B4_9000h
Init: _ccd7_lthree0_inst17_n122_aliasMSR: 23B4_A000h
Init: _ccd7_lthree0_inst18_n123_aliasMSR: 23B4_B000h
Init: _ccd7_lthree0_inst19_n124_aliasMSR: 23B4_C000h
Init: _ccd7_lthree0_inst20_n125_aliasMSR: 23B4_D000h
Init: _ccd7_lthree0_inst21_n126_aliasMSR: 23B4_E000h
Init: _ccd7_lthree0_inst22_n127_aliasMSR: 23B4_F000h
Init: _ccd8_lthree0_inst7_n128_aliasMSR: 2434_0000h
Init: _ccd8_lthree0_inst8_n129_aliasMSR: 2434_1000h
Init: _ccd8_lthree0_inst9_n130_aliasMSR: 2434_2000h
Init: _ccd8_lthree0_inst10_n131_aliasMSR: 2434_3000h
Init: _ccd8_lthree0_inst11_n132_aliasMSR: 2434_4000h
Init: _ccd8_lthree0_inst12_n133_aliasMSR: 2434_5000h
Init: _ccd8_lthree0_inst13_n134_aliasMSR: 2434_6000h
Init: _ccd8_lthree0_inst14_n135_aliasMSR: 2434_7000h
Init: _ccd8_lthree0_inst15_n136_aliasMSR: 2434_8000h
Init: _ccd8_lthree0_inst16_n137_aliasMSR: 2434_9000h
Init: _ccd8_lthree0_inst17_n138_aliasMSR: 2434_A000h
Init: _ccd8_lthree0_inst18_n139_aliasMSR: 2434_B000h
Init: _ccd8_lthree0_inst19_n140_aliasMSR: 2434_C000h
Init: _ccd8_lthree0_inst20_n141_aliasMSR: 2434_D000h
Init: _ccd8_lthree0_inst21_n142_aliasMSR: 2434_E000h
Init: _ccd8_lthree0_inst22_n143_aliasMSR: 2434_F000h
Init: _ccd9_lthree0_inst7_n144_aliasMSR: 24B4_0000h
Init: _ccd9_lthree0_inst8_n145_aliasMSR: 24B4_1000h
Init: _ccd9_lthree0_inst9_n146_aliasMSR: 24B4_2000h
Init: _ccd9_lthree0_inst10_n147_aliasMSR: 24B4_3000h
Init: _ccd9_lthree0_inst11_n148_aliasMSR: 24B4_4000h
Init: _ccd9_lthree0_inst12_n149_aliasMSR: 24B4_5000h
Init: _ccd9_lthree0_inst13_n150_aliasMSR: 24B4_6000h
Init: _ccd9_lthree0_inst14_n151_aliasMSR: 24B4_7000h
Init: _ccd9_lthree0_inst15_n152_aliasMSR: 24B4_8000h
Init: _ccd9_lthree0_inst16_n153_aliasMSR: 24B4_9000h
Init: _ccd9_lthree0_inst17_n154_aliasMSR: 24B4_A000h
Init: _ccd9_lthree0_inst18_n155_aliasMSR: 24B4_B000h
Init: _ccd9_lthree0_inst19_n156_aliasMSR: 24B4_C000h
Init: _ccd9_lthree0_inst20_n157_aliasMSR: 24B4_D000h
Init: _ccd9_lthree0_inst21_n158_aliasMSR: 24B4_E000h
Init: _ccd9_lthree0_inst22_n159_aliasMSR: 24B4_F000h
Init: _ccd10_lthree0_inst7_n160_aliasMSR: 2534_0000h
Init: _ccd10_lthree0_inst8_n161_aliasMSR: 2534_1000h
Init: _ccd10_lthree0_inst9_n162_aliasMSR: 2534_2000h
Init: _ccd10_lthree0_inst10_n163_aliasMSR: 2534_3000h
Init: _ccd10_lthree0_inst11_n164_aliasMSR: 2534_4000h
Init: _ccd10_lthree0_inst12_n165_aliasMSR: 2534_5000h
Init: _ccd10_lthree0_inst13_n166_aliasMSR: 2534_6000h
Init: _ccd10_lthree0_inst14_n167_aliasMSR: 2534_7000h
Init: _ccd10_lthree0_inst15_n168_aliasMSR: 2534_8000h
Init: _ccd10_lthree0_inst16_n169_aliasMSR: 2534_9000h
Init: _ccd10_lthree0_inst17_n170_aliasMSR: 2534_A000h
Init: _ccd10_lthree0_inst18_n171_aliasMSR: 2534_B000h

Init: _ccd10_lthree0_inst19_n172_aliasMSR: 2534_C000h
Init: _ccd10_lthree0_inst20_n173_aliasMSR: 2534_D000h
Init: _ccd10_lthree0_inst21_n174_aliasMSR: 2534_E000h
Init: _ccd10_lthree0_inst22_n175_aliasMSR: 2534_F000h
Init: _ccd11_lthree0_inst7_n176_aliasMSR: 25B4_0000h
Init: _ccd11_lthree0_inst8_n177_aliasMSR: 25B4_1000h
Init: _ccd11_lthree0_inst9_n178_aliasMSR: 25B4_2000h
Init: _ccd11_lthree0_inst10_n179_aliasMSR: 25B4_3000h
Init: _ccd11_lthree0_inst11_n180_aliasMSR: 25B4_4000h
Init: _ccd11_lthree0_inst12_n181_aliasMSR: 25B4_5000h
Init: _ccd11_lthree0_inst13_n182_aliasMSR: 25B4_6000h
Init: _ccd11_lthree0_inst14_n183_aliasMSR: 25B4_7000h
Init: _ccd11_lthree0_inst15_n184_aliasMSR: 25B4_8000h
Init: _ccd11_lthree0_inst16_n185_aliasMSR: 25B4_9000h
Init: _ccd11_lthree0_inst17_n186_aliasMSR: 25B4_A000h
Init: _ccd11_lthree0_inst18_n187_aliasMSR: 25B4_B000h
Init: _ccd11_lthree0_inst19_n188_aliasMSR: 25B4_C000h
Init: _ccd11_lthree0_inst20_n189_aliasMSR: 25B4_D000h
Init: _ccd11_lthree0_inst21_n190_aliasMSR: 25B4_E000h
Init: _ccd11_lthree0_inst22_n191_aliasMSR: 25B4_F000h

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](#)

_ccd[11:0]_lthree0_inst[15,7]_n[184,176,168,160,152,144,136,128,120,112,104,96,88,80,72,64,56,48,40,32,24,16,8,0]_aliasMSR; MSRC000_2076

_ccd[11:0]_lthree0_inst[16,8]_n[185,177,169,161,153,145,137,129,121,113,105,97,89,81,73,65,57,49,41,33,25,17,9,1]_aliasMSR; MSRC000_2086

_ccd[11:0]_lthree0_inst[17,9]_n[186,178,170,162,154,146,138,130,122,114,106,98,90,82,74,66,58,50,42,34,26,18,10,2]_aliasMSR; MSRC000_2096

_ccd[11:0]_lthree0_inst[18,10]_n[187,179,171,163,155,147,139,131,123,115,107,99,91,83,75,67,59,51,43,35,27,19,11,3]_aliasMSR; MSRC000_20A6

_ccd[11:0]_lthree0_inst[19,11]_n[188,180,172,164,156,148,140,132,124,116,108,100,92,84,76,68,60,52,44,36,28,20,12,4]_aliasMSR; MSRC000_20B6

_ccd[11:0]_lthree0_inst[20,12]_n[189,181,173,165,157,149,141,133,125,117,109,101,93,85,77,69,61,53,45,37,29,21,13,5]_aliasMSR; MSRC000_20C6

_ccd[11:0]_lthree0_inst[21,13]_n[190,182,174,166,158,150,142,134,126,118,110,102,94,86,78,70,62,54,46,38,30,22,14,6]_aliasMSR; MSRC000_20D6

_ccd[11:0]_lthree0_inst[22,14]_n[191,183,175,167,159,151,143,135,127,119,111,103,95,87,79,71,63,55,47,39,31,23,15,7]_aliasMSR; MSRC000_20E6

Bits	Description
63:32	Syndrom . 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	ErrorPriority . 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	Length . Read-write, Volatile . Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::L3::MCA_SYND_L3[Syndrome]. Length values greater than 32 (decimal) are interpreted as equal to 32 (decimal). 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	ErrorInformation . Read-write, Volatile . Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 72 [MCA_SYND_L3].

Table 72: 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]	Reserved Pack

	[7:0]	Reserved
Tag	[17:14]	Reserved
	[13]	Reserved
	[12]	Reserved
	[11:8]	Bank.
	[7:4]	Reserved
	[3:0]	Way
MultiHitTag	[17:0]	Reserved
dataArray	[17:14]	Reserved
	[13]	Reserved
	[12]	Reserved
	[11]	Reserved
	[10:8]	Bank[2:0]
	[7:4]	Reserved
	[3:0]	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.

_ccd[11:0]_lthree0_inst[15,7]_n[184,176,168,160,152,144,136,128,120,112,104,96,88,80,72,64,56,48,40,32,24,16,8,0]_aliasMSR; MSRC000_2078_ccd[11:0]_lthree0_inst[16,8]_n[185,177,169,161,153,145,137,129,121,113,105,97,89,81,73,65,57,49,41,33,25,17,9,1]_aliasMSR; MSRC000_2088_ccd[11:0]_lthree0_inst[17,9]_n[186,178,170,162,154,146,138,130,122,114,106,98,90,82,74,66,58,50,42,34,26,18,10,2]_aliasMSR; MSRC000_2098_ccd[11:0]_lthree0_inst[18,10]_n[187,179,171,163,155,147,139,131,123,115,107,99,91,83,75,67,59,51,43,35,27,19,11,3]_aliasMSR; MSRC000_20A8_ccd[11:0]_lthree0_inst[19,11]_n[188,180,172,164,156,148,140,132,124,116,108,100,92,84,76,68,60,52,44,36,28,20,12,4]_aliasMSR; MSRC000_20B8_ccd[11:0]_lthree0_inst[20,12]_n[189,181,173,165,157,149,141,133,125,117,109,101,93,85,77,69,61,53,45,37,29,21,13,5]_aliasMSR; MSRC000_20C8_ccd[11:0]_lthree0_inst[21,13]_n[190,182,174,166,158,150,142,134,126,118,110,102,94,86,78,70,62,54,46,38,30,22,14,6]_aliasMSR; MSRC000_20D8_ccd[11:0]_lthree0_inst[22,14]_n[191,183,175,167,159,151,143,135,127,119,111,103,95,87,79,71,63,55,47,39,31,23,15,7]_aliasMSR; MSRC000_20E8

Bits	Description
63	Val. Read-write, Volatile . Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	Overflow. 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 overwrite priorities.)
61:59	RESERV4. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	AddrV. Read-write, Volatile . Reset: Cold,0. 1=MCA::L3::MCA_DEADDR_L3 contains address information associated with the error.
57:54	RESERV3. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	SyndV. 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	RESERV2. Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	Deferred. 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	RESERV1. Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	AddrLsb. 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	RESERV0. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	ErrorCodeExt. 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	ErrorCode. 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].

_ccd[11:0]_lthree0_inst[15,7]_n[184,176,168,160,152,144,136,128,120,112,104,96,88,80,72,64,56,48,40,32,24,16,8,0]_aliasMSR; MSRC000_2079

_ccd[11:0]_lthree0_inst[16,8]_n[185,177,169,161,153,145,137,129,121,113,105,97,89,81,73,65,57,49,41,33,25,17,9,1]_aliasMSR; MSRC000_2089

_ccd[11:0]_lthree0_inst[17,9]_n[186,178,170,162,154,146,138,130,122,114,106,98,90,82,74,66,58,50,42,34,26,18,10,2]_aliasMSR; MSRC000_2099

_ccd[11:0]_lthree0_inst[18,10]_n[187,179,171,163,155,147,139,131,123,115,107,99,91,83,75,67,59,51,43,35,27,19,11,3]_aliasMSR; MSRC000_20A9

_ccd[11:0]_lthree0_inst[19,11]_n[188,180,172,164,156,148,140,132,124,116,108,100,92,84,76,68,60,52,44,36,28,20,12,4]_aliasMSR; MSRC000_20B9

_ccd[11:0]_lthree0_inst[20,12]_n[189,181,173,165,157,149,141,133,125,117,109,101,93,85,77,69,61,53,45,37,29,21,13,5]_aliasMSR; MSRC000_20C9

_ccd[11:0]_lthree0_inst[21,13]_n[190,182,174,166,158,150,142,134,126,118,110,102,94,86,78,70,62,54,46,38,30,22,14,6]_aliasMSR; MSRC000_20D9

_ccd[11:0]_lthree0_inst[22,14]_n[191,183,175,167,159,151,143,135,127,119,111,103,95,87,79,71,63,55,47,39,31,23,15,7]_aliasMSR; MSRC000_20E9

Bits	Description
63:0	ErrorAddr. 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.

_ccd[11:0]_lthree0_inst[15,7]_n[184,176,168,160,152,144,136,128,120,112,104,96,88,80,72,64,56,48,40,32,24,16,8,0]_aliasMSR; MSRC001_0407

_ccd[11:0]_lthree0_inst[16,8]_n[185,177,169,161,153,145,137,129,121,113,105,97,89,81,73,65,57,49,41,33,25,17,9,1]_aliasMSR; MSRC001_0408

_ccd[11:0]_lthree0_inst[17,9]_n[186,178,170,162,154,146,138,130,122,114,106,98,90,82,74,66,58,50,42,34,26,18,10,2]_aliasMSR; MSRC001_0409

_ccd[11:0]_lthree0_inst[18,10]_n[187,179,171,163,155,147,139,131,123,115,107,99,91,83,75,67,59,51,43,35,27,19,11,3]_aliasMSR; MSRC001_040A

_ccd[11:0]_lthree0_inst[19,11]_n[188,180,172,164,156,148,140,132,124,116,108,100,92,84,76,68,60,52,44,36,28,20,12,4]_aliasMSR; MSRC001_040B

_ccd[11:0]_lthree0_inst[20,12]_n[189,181,173,165,157,149,141,133,125,117,109,101,93,85,77,69,61,53,45,37,29,21,13,5]_aliasMSR; MSRC001_040C

_ccd[11:0]_lthree0_inst[21,13]_n[190,182,174,166,158,150,142,134,126,118,110,102,94,86,78,70,62,54,46,38,30,22,14,6]_aliasMSR; MSRC001_040D

_ccd[11:0]_lthree0_inst[22,14]_n[191,183,175,167,159,151,143,135,127,119,111,103,95,87,79,71,63,55,47,39,31,23,15,7]_aliasMSR; MSRC001_040E

Bits	Description
63:10	Reserved.
9	DsmMce. Read-write. Reset: 0. Machine check error initiated by DSM action
8	XiWcbParityPoison. Read-write. Reset: 0. Xi Wcb Parity Poison Creation Event
7	Hwa. Read-write. Reset: 0. L3 Hardware Assertion.
6	XiVictimQueue. Read-write. Reset: 0. L3 Victim Queue Data Fabric Error.
5	SdpParity. Read-write. Reset: 0. SDP Parity Error from XI.
4	DataArray. Read-write. Reset: 0. L3M Data ECC Error.
3	MultiHitTag. Read-write. Reset: 0. L3M Tag Multi-way-hit Error.
2	Tag. Read-write. Reset: 0. L3M Tag ECC Error.
1	MultiHitShadowTag. Read-write. Reset: 0. Shadow Tag Macro Multi-way-hit Error.
0	ShadowTag. 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](#)[_ccd\[11:0\]_lthree0_inst\[15,7\]_n\[184,176,168,160,152,144,136,128,120,112,104,96,88,80,72,64,56,48,40,32,24,16,8,0\]_aliasMSR; MSRC000_207E](#)[_ccd\[11:0\]_lthree0_inst\[16,8\]_n\[185,177,169,161,153,145,137,129,121,113,105,97,89,81,73,65,57,49,41,33,25,17,9,1\]_aliasMSR; MSRC000_208E](#)[_ccd\[11:0\]_lthree0_inst\[17,9\]_n\[186,178,170,162,154,146,138,130,122,114,106,98,90,82,74,66,58,50,42,34,26,18,10,2\]_aliasMSR; MSRC000_209E](#)[_ccd\[11:0\]_lthree0_inst\[18,10\]_n\[187,179,171,163,155,147,139,131,123,115,107,99,91,83,75,67,59,51,43,35,27,19,11,3\]_aliasMSR; MSRC000_20AE](#)[_ccd\[11:0\]_lthree0_inst\[19,11\]_n\[188,180,172,164,156,148,140,132,124,116,108,100,92,84,76,68,60,52,44,36,28,20,12,4\]_aliasMSR; MSRC000_20BE](#)[_ccd\[11:0\]_lthree0_inst\[20,12\]_n\[189,181,173,165,157,149,141,133,125,117,109,101,93,85,77,69,61,53,45,37,29,21,13,5\]_aliasMSR; MSRC000_20CE](#)[_ccd\[11:0\]_lthree0_inst\[21,13\]_n\[190,182,174,166,158,150,142,134,126,118,110,102,94,86,78,70,62,54,46,38,30,22,14,6\]_aliasMSR; MSRC000_20DE](#)[_ccd\[11:0\]_lthree0_inst\[22,14\]_n\[191,183,175,167,159,151,143,135,127,119,111,103,95,87,79,71,63,55,47,39,31,23,15,7\]_aliasMSR; MSRC000_20EE](#)

Bits	Description
63:0	Syndrome. 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](#)[_ccd\[11:0\]_lthree0_inst\[15,7\]_n\[184,176,168,160,152,144,136,128,120,112,104,96,88,80,72,64,56,48,40,32,24,16,8,0\]_aliasMSR; MSRC000_207F](#)[_ccd\[11:0\]_lthree0_inst\[16,8\]_n\[185,177,169,161,153,145,137,129,121,113,105,97,89,81,73,65,57,49,41,33,25,17,9,1\]_aliasMSR; MSRC000_208F](#)[_ccd\[11:0\]_lthree0_inst\[17,9\]_n\[186,178,170,162,154,146,138,130,122,114,106,98,90,82,74,66,58,50,42,34,26,18,10,2\]_aliasMSR; MSRC000_209F](#)[_ccd\[11:0\]_lthree0_inst\[18,10\]_n\[187,179,171,163,155,147,139,131,123,115,107,99,91,83,75,67,59,51,43,35,27,19,11,3\]_aliasMSR; MSRC000_20AF](#)[_ccd\[11:0\]_lthree0_inst\[19,11\]_n\[188,180,172,164,156,148,140,132,124,116,108,100,92,84,76,68,60,52,44,36,28,20,12,4\]_aliasMSR; MSRC000_20BF](#)[_ccd\[11:0\]_lthree0_inst\[20,12\]_n\[189,181,173,165,157,149,141,133,125,117,109,101,93,85,77,69,61,53,45,37,29,21,13,5\]_aliasMSR; MSRC000_20CF](#)[_ccd\[11:0\]_lthree0_inst\[21,13\]_n\[190,182,174,166,158,150,142,134,126,118,110,102,94,86,78,70,62,54,46,38,30,22,14,6\]_aliasMSR; MSRC000_20DF](#)[_ccd\[11:0\]_lthree0_inst\[22,14\]_n\[191,183,175,167,159,151,143,135,127,119,111,103,95,87,79,71,63,55,47,39,31,23,15,7\]_aliasMSR; MSRC000_20EF](#)

Bits	Description
63:0	Syndrome. 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

MSR0000_045C...MSRC000_2180 [CS Machine Check Control] (MCA::CS::MCA_CTL_CS)

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.	
_inst[CS[8,6,4,2,10,0],CMP[2,0]]_n[14,12,10,8,6,4,2,0]_aliasMSRLEGACY; MSR0000_045C	
_inst[CS[9,7,5,3,11,1],CMP[3,1]]_n[15,13,11,9,7,5,3,1]_aliasMSRLEGACY; MSR0000_0460	
_inst[CS[8,6,4,2,10,0],CMP[2,0]]_n[14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2170	
_inst[CS[9,7,5,3,11,1],CMP[3,1]]_n[15,13,11,9,7,5,3,1]_aliasMSR; MSRC000_2180	
Bits	Description
63:21	Reserved.
20	ST_TXN_ERR. Read-write. Reset: 0. Shadow Tag Transaction Error: An illegal Shadow Tag access occurred.
19	ST_ECC_ERR. Read-write. Reset: 0. Shadow Tag ECC Error: An ECC error occurred on a shadow tag array access.
18	ST_PRT_ERR. Read-write. Reset: 0. Shadow Tag Array Protocol Error: Indicates a Cache Coherence Issue.
17	HWA. Read-write. Reset: 0. Hardware Assert Error.
16	FTI_ND_SEC_VIOL. 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	FTI_ND_ADDR_VIOL. 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	FTI_ND_ILL_REQ. Read-write. Reset: 0. Illegal Request: An illegal request was received from the transport layer on the request no data channel.
13	CNTR_UNFL. Read-write. Reset: 0. Counter underflow error.
12	CNTR_OVFL. Read-write. Reset: 0. Counter overflow error.
11	SDP_UNEXP_RETRY. Read-write. Reset: 0. SDP read response had an unexpected RETRY error.
10	SPF_ECC_ERR. Read-write. Reset: 0. Probe Filter ECC Error: An ECC error occurred on a probe filter access.
9	SPF_PROTOCOL_ERR. Read-write. Reset: 0. Probe Filter Protocol Error: Indicates a Cache Coherence Issue.
8	SDP_RSP_NO_MTCH. Read-write. Reset: 0. SDP read response had no match in the CS queue.
7	ATM_PAR_ERR. Read-write. Reset: 0. Atomic Request Parity Error: Parity error on read of an atomic transaction.
6	SDP_PAR_ERR. Read-write. Reset: 0. Read Response Parity Error: Parity error on incoming read response data.
5	FTI_PAR_ERR. Read-write. Reset: 0. Request or Probe Parity Error: Parity error on incoming request or probe response data.
4	FTI_RSP_NO_MTCH. Read-write. Reset: 0. Unexpected Response: A response was received from the transport layer which does not match any request.
3	FTI_ILL_RSP. Read-write. Reset: 0. Illegal Response: An illegal response was received from the transport layer.
2	FTI_SEC_VIOL. 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	FTI_ADDR_VIOL. 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	FTI_ILL_REQ. Read-write. Reset: 0. Illegal Request: An illegal request was received from the transport layer on the primary request channel.

MSR0000_045D...MSRC000_2181 [CS Machine Check Status] (MCA::CS::MCA_STATUS_CS)

Reset: Cold,0000_0000_0000_0000h.

Logs information associated with errors.

_inst[CS[8,6,4,2,10,0],CMP[2,0]]_n[14,12,10,8,6,4,2,0] aliasMSRLEGACY; MSR0000_045D_inst[CS[9,7,5,3,11,1],CMP[3,1]]_n[15,13,11,9,7,5,3,1] aliasMSRLEGACY; MSR0000_0461_inst[CS[8,6,4,2,10,0],CMP[2,0]]_n[14,12,10,8,6,4,2,0] aliasMSR; MSRC000_2171_inst[CS[9,7,5,3,11,1],CMP[3,1]]_n[15,13,11,9,7,5,3,1] aliasMSR; MSRC000_2181

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. 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	MiscV. 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	AddrV. 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	PCC. 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	ErrCoreIdVal. 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	TCC. 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	RESERV54. 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	SyndV. 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	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. 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	RESERV22. 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	ErrorCodeExt. 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	ErrorCode. 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 73: 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_NO_MTCH	0x4	1	1	1	0	0	0
FTI_PAR_ERR	0x5	0	0	0	1	0	1

SDP_PAR_ERR	0x6	0	0	0	1	0	1
ATM_PAR_ERR	0x7	0	0	0	1	0	1
SDP_RSP_NO_MTCH	0x8	1	1	1	0	0	0
SPF_PROT_OCOL_ERR	0x9	1	1	1	0	0	0
SPF_ECC_ERR	0xa	0/1	0/1	0/1	0	0	1
SDP_UNEXP_RETRY	0xb	1	1	1	0	0	1
CNTR_OVFL	0xc	1	1	1	0	0	0
CNTR_UNFL	0xd	1	1	1	0	0	0
FTI_ND_ILL_REQ	0xe	0	0	0	1	0	1
FTI_ND_ADDR_VIOL	0xf	0	0	0	1	0	1
FTI_ND_SEC_VIOL	0x10	0	0	0	1	0	1
HWA	0x11	1	1	1	0	0	0
ST_PRT_ERR	0x12	1	1	1	0	0	0
ST_ECC_ERR	0x13	0/1	0/1	0/1	0	0	0
ST_TXN_ERR	0x14	1	1	1	0	0	0

MSR0000_045E...MSRC000_2182 [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.

_inst[CS[8,6,4,2,10],CMP[2,0]]_n[14,12,10,8,6,4,2,0]_aliasMSRLEGACY; MSR0000_045E

_inst[CS[9,7,5,3,11,1],CMP[3,1]]_n[15,13,11,9,7,5,3,1]_aliasMSRLEGACY; MSR0000_0462

_inst[CS[8,6,4,2,10],CMP[2,0]]_n[14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2172

_inst[CS[9,7,5,3,11,1],CMP[3,1]]_n[15,13,11,9,7,5,3,1]_aliasMSR; MSRC000_2182

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::CS::MCA_STATUS_CS. For physical addresses, the most significant bit is given by Core::X86::Cpuid::LongModeInfo[PhysAddrSize].

Table 74: MCA_ADDR_CS

Error Type	Bits	Description
FTI_ILL_REQ	[55:2]	Address
FTI_ADDR_VIOL	[55:2]	Address
FTI_SEC_VIOL	[55:2]	Address
FTI_ILL_RSP	[63:0]	Reserved
FTI_RSP_NO_MTCH	[63:0]	Reserved

FTI_PAR_ERR	[55:2]	Address
SDP_PAR_ERR	[55:2]	Address
ATM_PAR_ERR	[55:2]	Address
SDP_RSP_NO_MTCH	[63:0]	Reserved
SPF_PROTOCOL_ERR	[63:0]	Reserved
SPF_ECC_ERR	[55:2]	Address
SDP_UNEXP_RETRY	[55:2]	Address
CNTR_OVFL	[63:0]	Reserved
CNTR_UNFL	[63:0]	Reserved
FTI_ND_ILL_REQ	[55:2]	Address
FTI_ND_ADDR_VIOL	[55:2]	Address
FTI_ND_SEC_VIOL	[55:2]	Address
HWA	[55:2]	Address
ST_PRT_ERR	[63:0]	Reserved
ST_ECC_ERR	[63:0]	Reserved
ST_TXN_ERR	[63:0]	Reserved

MSR0000_045F...MSRC000_2183 [CS Machine Check Miscellaneous 0] (MCA::CS::MCA_MISC0_CS)

Log miscellaneous information associated with errors.

[_inst\[CS\[8,6,4,2,10,0\],CMP\[2,0\]\]_n\[14,12,10,8,6,4,2,0\]_aliasMSRLEGACY; MSR0000_045F](#)[_inst\[CS\[9,7,5,3,11,1\],CMP\[3,1\]\]_n\[15,13,11,9,7,5,3,1\]_aliasMSRLEGACY; MSR0000_0463](#)[_inst\[CS\[8,6,4,2,10,0\],CMP\[2,0\]\]_n\[14,12,10,8,6,4,2,0\]_aliasMSR; MSRC000_2173](#)[_inst\[CS\[9,7,5,3,11,1\],CMP\[3,1\]\]_n\[15,13,11,9,7,5,3,1\]_aliasMSR; MSRC000_2183](#)

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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::CS::MCA_MISC0_CS[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	LvtOffset. 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::CS::MCA_MISC0_CS[Locked]) ? Read-write : Read-only.
51	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msrr::HWCR[McStatusWrEn] !MCA::CS::MCA_MISC0_CS[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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::CS::MCA_MISC0_CS[Locked]) ? Read-write : Read-only.
48	Ovrflw. 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::CS::MCA_MISC0_CS[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	ErrCnt. 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::CS::MCA_MISC0_CS[Locked]) ? Read-write : Read-only.
31:24	BlkPtr. Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

MSRC000_21[7...8]4 [CS Machine Check Configuration] (MCA::CS::MCA_CONFIG_CS)

Reset: 0000_0000_0000_0125h.

Controls configuration of the associated machine check bank.

_inst[CS[8,6,4,2,10,0],CMP[2,0]]_n[14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2174_inst[CS[9,7,5,3,11,1],CMP[3,1]]_n[15,13,11,9,7,5,3,1]_aliasMSR; MSRC000_2184

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	LogDeferredInMcaStat. 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	McaXEnable. 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	IntPresent. 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	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::CS::MCA_CONFIG_CS[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. 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	DeferredErrorLoggingSupported. 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	McaX. 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[7...8]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.

_inst[CS[8,6,4,2,10,0],CMP[2,0]]_n[14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2175

_inst[CS[9,7,5,3,11,1],CMP[3,1]]_n[15,13,11,9,7,5,3,1]_aliasMSR; MSRC000_2185

Bits	Description
63:48	McaType . Read-only. Reset: 0002h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi . 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	HardwareID . Read-only. Reset: 02Eh. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _instCMP0_n12_aliasMSR: 1F00_0C00h
	Init: _instCMP1_n13_aliasMSR: 1F00_0D00h
	Init: _instCMP2_n14_aliasMSR: 1F00_0E00h
	Init: _instCMP3_n15_aliasMSR: 1F00_0F00h
	Init: _instCS0_n0_aliasMSR: 1F00_0000h
	Init: _instCS1_n1_aliasMSR: 1F00_0100h
	Init: _instCS2_n2_aliasMSR: 1F00_0200h
	Init: _instCS3_n3_aliasMSR: 1F00_0300h
	Init: _instCS4_n4_aliasMSR: 1F00_0400h
	Init: _instCS5_n5_aliasMSR: 1F00_0500h
	Init: _instCS6_n6_aliasMSR: 1F00_0600h
	Init: _instCS7_n7_aliasMSR: 1F00_0700h
	Init: _instCS8_n8_aliasMSR: 1F00_0800h
	Init: _instCS9_n9_aliasMSR: 1F00_0900h
	Init: _instCS10_n10_aliasMSR: 1F00_0A00h
	Init: _instCS11_n11_aliasMSR: 1F00_0B00h

MSRC000_21[7...8]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](#)

_inst[CS[8,6,4,2,10,0],CMP[2,0]]_n[14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2176

_inst[CS[9,7,5,3,11,1],CMP[3,1]]_n[15,13,11,9,7,5,3,1]_aliasMSR; MSRC000_2186

Bits	Description
63:32	Syndrom . 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	ErrorPriority . 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	Length . Read-write, Volatile . Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::CS::MCA_SYND_CS[Syndrome]. Length values greater than 32 (decimal) are interpreted as equal to 32 (decimal). 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	ErrorInformation . Read-write, Volatile . Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 75 [MCA_SYND_CS].

Table 75: 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_PROTOCOL_ERR	[36:32]	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[7...8]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.

_inst[CS[8,6,4,2,10,0],CMP[2,0]]_n[14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2178_inst[CS[9,7,5,3,11,1],CMP[3,1]]_n[15,13,11,9,7,5,3,1]_aliasMSR; MSRC000_2188

Bits	Description
63	Val. Read-write, Volatile . Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	Overflow. 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 overwrite priorities.)
61:59	RESERV4. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	AddrV. Read-write, Volatile . Reset: Cold,0. 1=MCA::CS::MCA_DEADDR_CS contains address information associated with the error.
57:54	RESERV3. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	SyndV. 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	RESERV2. Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	Deferred. 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	RESERV1. Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	AddrLsb. 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	RESERV0. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	ErrorCodeExt. 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	ErrorCode. Read-write, Volatile . Reset: Cold,0000h. Error code for this error.

MSRC000_21[7...8]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].

_inst[CS[8,6,4,2,10,0],CMP[2,0]]_n[14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2179_inst[CS[9,7,5,3,11,1],CMP[3,1]]_n[15,13,11,9,7,5,3,1]_aliasMSR; MSRC000_2189

Bits	Description
63:0	ErrorAddr. 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[7...8] [CS Machine Check Control Mask] (MCA::CS::MCA_CTL_MASK_CS)

Read-write. Reset: 0000_0000_0000_0000h.

Inhibit detection of an error source.

`_inst[CS[8,6,4,2,10,0],CMP[2,0]]_n[14,12,10,8,6,4,2,0]_aliasMSR; MSRC001_0417``_inst[CS[9,7,5,3,11,1],CMP[3,1]]_n[15,13,11,9,7,5,3,1]_aliasMSR; MSRC001_0418`

Bits	Description
63:21	Reserved.
20	ST_TXN_ERR. Read-write. Reset: 0. Shadow Tag Transaction Error: An illegal Shadow Tag access occurred.
19	ST_ECC_ERR. Read-write. Reset: 0. Shadow Tag ECC Error: An ECC error occurred on a shadow tag array access.
18	ST_PRT_ERR. Read-write. Reset: 0. Shadow Tag Array Protocol Error: Indicates a Cache Coherence Issue.
17	HWA. Read-write. Reset: 0. Hardware Assert Error.
16	FTI_ND_SEC_VIOL. 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	FTI_ND_ADDR_VIOL. 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	FTI_ND_ILL_REQ. Read-write. Reset: 0. Illegal Request: An illegal request was received from the transport layer on the request no data channel.
13	CNTR_UNFL. Read-write. Reset: 0. Counter underflow error.
12	CNTR_OVFL. Read-write. Reset: 0. Counter overflow error.
11	SDP_UNEXP_RETRY. Read-write. Reset: 0. SDP read response had an unexpected RETRY error.
10	SPF_ECC_ERR. Read-write. Reset: 0. Probe Filter ECC Error: An ECC error occurred on a probe filter access.
9	SPF_PROTOCOL_ERR. Read-write. Reset: 0. Probe Filter Protocol Error: Indicates a Cache Coherence Issue.
8	SDP_RSP_NO_MTCH. Read-write. Reset: 0. SDP read response had no match in the CS queue.
7	ATM_PAR_ERR. Read-write. Reset: 0. Atomic Request Parity Error: Parity error on read of an atomic transaction.
6	SDP_PAR_ERR. Read-write. Reset: 0. Read Response Parity Error: Parity error on incoming read response data.
5	FTI_PAR_ERR. Read-write. Reset: 0. Request or Probe Parity Error: Parity error on incoming request or probe response data.
4	FTI_RSP_NO_MTCH. Read-write. Reset: 0. Unexpected Response: A response was received from the transport layer which does not match any request.
3	FTI_ILL_RSP. Read-write. Reset: 0. Illegal Response: An illegal response was received from the transport layer.
2	FTI_SEC_VIOL. 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	FTI_ADDR_VIOL. 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	FTI_ILL_REQ. Read-write. Reset: 0. Illegal Request: An illegal request was received from the transport layer on the primary request channel.

MSRC000_21[7...8]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](#)`_inst[CS[8,6,4,2,10,0],CMP[2,0]]_n[14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_217E``_inst[CS[9,7,5,3,11,1],CMP[3,1]]_n[15,13,11,9,7,5,3,1]_aliasMSR; MSRC000_218E`

Bits	Description
63:0	Syndrome. 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[7...8]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](#)

_inst[CS[8,6,4,2,10,0],CMP[2,0]]_n[14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_217F

_inst[CS[9,7,5,3,11,1],CMP[3,1]]_n[15,13,11,9,7,5,3,1]_aliasMSR; MSRC000_218F

Bits	Description
63:0	Syndrone. 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_0478...MSRC000_21E0 [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_0478

_instPIE0_n0_aliasMSR; MSRC000_21E0

Bits	Description
63:9	Reserved.
8	DSM_ACT. Read-write. Reset: 0.
7	RSLVFCI. Read-write. Reset: 0. Register access during DF Cstate: Register access is detected from SMN during DF Cstate.
6	CNLI. Read-write. Reset: 0. An SRAM ECC error in the CNLI block.
5	WDT. Read-write. Reset: 0. Watch Dog Timer: A watch dog timer expired.
4	DEF. Read-write. Reset: 0. A deferred error was detected in the DF .
3	FTL_DAT_STAT. Read-write. Reset: 0. Poison data consumption: Poison data was written to an internal PIE register.
2	GMI. Read-write. Reset: 0. Reserved (Was used GMI link errors).
1	CSW. Read-write. Reset: 0. Register security violation: A security violation was detected on an access to an internal PIE register.
0	HW_ASSERT. Read-write. Reset: 0. Hardware Assert: A hardware assert was detected.

MSR0000_0479...MSRC000_21E1 [PIE Machine Check Status] (MCA::PIE::MCA_STATUS_PIE)

Reset: Cold,0000_0000_0000_0000h.

Logs information associated with errors.

_instPIE0_n0_aliasMSRLEGACY; MSR0000_0479

_instPIE0_n0_aliasMSR; MSRC000_21E1

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. 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	MiscV. 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	AddrV. 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	PCC. 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	ErrCoreIdVal. 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	TCC. 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	RESERV54. 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	SyndV. 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	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. 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	RESERV22. 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	ErrorCodeExt. 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	ErrorCode. 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 76: 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_047A...MSRC000_21E2 [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_047A

_instPIE0_n0_aliasMSR; MSRC000_21E2

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::PIE::MCA_STATUS_PIE. For physical addresses, the most significant bit is given by Core::X86::Cpuid::LongModeInfo[PhysAddrSize].

Table 77: 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_047B...MSRC000_21E3 [PIE Machine Check Miscellaneous 0] (MCA::PIE::MCA_MISC0_PIE)

Log miscellaneous information associated with errors.

_instPIE0_n0_aliasMSRLEGACY; MSR0000_047B

_instPIE0_n0_aliasMSR; MSRC000_21E3

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr:: HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr:: HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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	LvtOffset. 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	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr:: HWCR[McStatusWrEn] !MCA::PIE::MCA_MISC0_PIE[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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	Ovrflw. 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	ErrCnt. 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	BlkPtr. Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

MSRC000_21E4 [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_21E4

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	LogDeferredInMcaStat. 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	McaXEnable. 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	IntPresent. 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	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::PIE::MCA_CONFIG_PIE[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. 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	DeferredErrorLoggingSupported. 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	McaX. 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_21E5 [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_21E5

Bits	Description
63:48	McaType . Read-only. Reset: 0001h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi . 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	HardwareID . Read-only. Reset: 02Eh. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId . Read-write. Reset: 0000_0000h. Init: 1F00_3400h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.

MSRC000_21E6 [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_21E6

Bits	Description
63:32	Syndrom . 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	ErrorPriority . 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	Length . Read-write, Volatile . Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::PIE::MCA_SYND_PIE[Syndrome]. Length values greater than 32 (decimal) are interpreted as equal to 32 (decimal). 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	ErrorInformation . Read-write, Volatile . Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 78 [MCA_SYND_PIE].

Table 78: 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_21E8 [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_21E8

Bits	Description
63	Val. Read-write, Volatile . Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	Overflow. 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 overwrite priorities.)
61:59	RESERV4. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	AddrV. Read-write, Volatile . Reset: Cold,0. 1=MCA::PIE::MCA_DEADDR_PIE contains address information associated with the error.
57:54	RESERV3. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	SyndV. 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	RESERV2. Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	Deferred. 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	RESERV1. Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	AddrLsb. 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	RESERV0. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	ErrorCodeExt. 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	ErrorCode. Read-write, Volatile . Reset: Cold,0000h. Error code for this error.

MSRC000_21E9 [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_21E9

Bits	Description
63:0	ErrorAddr. 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_041E [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_041E

Bits	Description
63:9	Reserved.
8	DSM_ACT. Read-write. Reset: 0.
7	RSLVFCI. Read-write. Reset: 0. Register access during DF Cstate: Register access is detected from SMN during DF Cstate.
6	CNLI. Read-write. Reset: 0. An SRAM ECC error in the CNLI block.
5	WDT. Read-write. Reset: 0. Watch Dog Timer: A watch dog timer expired.
4	DEF. Read-write. Reset: 0. A deferred error was detected in the DF .
3	FTI_DAT_STAT. Read-write. Reset: 0. Poison data consumption: Poison data was written to an internal PIE register.
2	GMI. Read-write. Reset: 0. Reserved (Was used GMI link errors).
1	CSW. Read-write. Reset: 0. Register security violation: A security violation was detected on an access to an internal PIE register.
0	HW_ASSERT. Read-write. Reset: 0. Hardware Assert: A hardware assert was detected.

MSRC000_21EE [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_21EE

Bits	Description
63:0	Syndrome. 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_21EF [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_21EF

Bits	Description
63:0	Syndrome. 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_0454...MSRC000_2160 [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.	
<u>_inst[UMCWPHY[8,6,4,2,10,0]UMC]_n[10,8,6,4,2,0]_umc0_aliasMSRLEGACY; MSR0000_0454</u>	
<u>_inst[UMCWPHY[9,7,5,3,1,11]UMC]_n[11,9,7,5,3,1]_umc0_aliasMSRLEGACY; MSR0000_0458</u>	
<u>_inst[UMCWPHY[8,6,4,2,10,0]UMC]_n[10,8,6,4,2,0]_umc0_aliasMSR; MSRC000_2150</u>	
<u>_inst[UMCWPHY[9,7,5,3,1,11]UMC]_n[11,9,7,5,3,1]_umc0_aliasMSR; MSRC000_2160</u>	
Bits	Description
63:17	Reserved.
16	RfmSramEccErr. Read-write. Reset: 0. RFM SRAM ECC error. An ECC error occurred on a RFM SRAM in the processor.
15:12	Reserved.
11	RdCrcErr. Read-write. Reset: 0. Read CRC error. CRC error occurred on a DRAM read from any subchannel
10	ThrttlErr. Read-write. Reset: 0. Indicates that UMC is throttling
9	EcsErr. Read-write. Reset: 0. ECS Error. Indicates that a device exceeded the ECS Error Threshold Count.
8	EcsRowErr. Read-write. Reset: 0. ECS Row Error. Indicates that a single device row exceeded four code word errors.
7	AesSramEccErr. Read-write. Reset: 0. AES SRAM ECC error. An ECC error occurred on a AES SRAM in the processor.
6	DcqSramEccErr. Read-write. Reset: 0. DCQ SRAM ECC error. An ECC error occurred on a DCQ SRAM in the processor.
5	WriteDataCrcErr. Read-write. Reset: 0. Write data CRC error. A write data CRC error occurred on the DRAM data bus.
4	AddressCommandParityErr. Read-write. Reset: 0. Address/Command parity error. A parity error occurred on the DRAM address/command bus.
3	ApbErr. Read-write. Reset: 0. Advanced peripheral bus error. An error occurred on the advanced peripheral bus.
2	SdpParityErr. Read-write. Reset: 0. SDP parity error. A parity error was detected on write data from the data fabric in the processor.
1	WriteDataPoisonErr. 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	DramEccErr. Read-write. Reset: 0. DRAM ECC error. An ECC error occurred on a DRAM read.

MSR0000_0455...MSRC000_2161 [UMC Machine Check Status] (MCA::UMC::MCA_STATUS_UMC)

Reset: Cold,0000_0000_0000_0000h.

Logs information associated with errors.

_inst[UMCWPHY[8,6,4,2,10,0]UMC]_n[10,8,6,4,2,0]_umc0_aliasMSRLEGACY; MSR0000_0455

_inst[UMCWPHY[9,7,5,3,1,11]UMC]_n[11,9,7,5,3,1]_umc0_aliasMSRLEGACY; MSR0000_0459

_inst[UMCWPHY[8,6,4,2,10,0]UMC]_n[10,8,6,4,2,0]_umc0_aliasMSR; MSRC000_2151

_inst[UMCWPHY[9,7,5,3,1,11]UMC]_n[11,9,7,5,3,1]_umc0_aliasMSR; MSRC000_2161

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. 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	MiscV. 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	AddrV. 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	PCC. 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	ErrCoreIdVal. 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	TCC. 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	RESERV54. 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	SyndV. 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	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. 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::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
23:22	RESERV22. 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	ErrorCodeExt. 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::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
15:0	ErrorCode. 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 79: 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
AddressCommandParityErr	0x4	0/1	0/1	0/1	0	0	0/1

WriteDataCr cErr	0x5	0/1	0/1	0/1	0	0	0/1
DcqSramEcc Err	0x6	0/1	0/1	0/1	0	0	0
AesSramEcc Err	0x7	0/1	0/1	0/1	0	0	0
EcsRowErr	0x8	0	0	0	0	0	0
EcsErr	0x9	0	0	0	0	0	0
ThrttlErr	0xa	0	0	0	0	0	0
RdCrcErr	0xb	0	0	0	1	0	1
Unused0	0xc					0	0
Unused1	0xd					0	0
Unused2	0xe					0	0
Unused3	0xf					0	0
RfmSramEc cErr	0x10	0/1	0/1	0/1	0	0	0

MSR0000_0456...MSRC000_2162 [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.

_inst[UMCWPHY[8,6,4,2,10,0]UMC]_n[10,8,6,4,2,0]_umc0_aliasMSRLEGACY; MSR0000_0456

_inst[UMCWPHY[9,7,5,3,1,11]UMC]_n[11,9,7,5,3,1]_umc0_aliasMSRLEGACY; MSR0000_045A

_inst[UMCWPHY[8,6,4,2,10,0]UMC]_n[10,8,6,4,2,0]_umc0_aliasMSR; MSRC000_2152

_inst[UMCWPHY[9,7,5,3,1,11]UMC]_n[11,9,7,5,3,1]_umc0_aliasMSR; MSRC000_2162

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::UMC::MCA_STATUS_UMC. For physical addresses, the most significant bit is given by Core::X86::Cpuid::LongModeInfo[PhysAddrSize].

Table 80: MCA_ADDR_UMC

Error Type	Bits	Description
DramEccErr	[55:39] [39:4]	Reserved Reserved
WriteDataPoisonErr	[55:0]	Reserved
SdpParityErr	[55:0]	Reserved
ApbErr	[55:30] [29:0]	Reserved Reserved
AddressCommandParityErr	[55:42] [41] [40] [39:0]	Reserved 0 = SubChannel A. 1 = SubChannel B 0 = DIMM0. 1 = DIMM1. RCD error log control words 24:20
WriteDataCrcErr	[55:38] [37:36] [35:32] [31:0]	Reserved Reserved Chip Select Reserved
DcqSramEccErr	[55:0]	Reserved
AesSramEccErr	[55:0]	Reserved
EcsRowErr	[55:23]	Reserved

	[22:20] [19:18] [17:0]	DDR5 BG[2:0] DDR5 BA[1:0] DDR5 ROW (This information is derived from MR18, MR17, MR16.)
EcsErr	[55:23] [22:20] [19:18] [17:0]	Reserved DDR5 BG[2:0] DDR5 BA[1:0] DDR5 ROW (This information is derived from MR18, MR17, MR16.)
RdCrcErr	[55:39] [39:4]	Reserved Reserved

MSR0000_0457...MSRC000_2163 [UMC Machine Check Miscellaneous 0] (MCA::UMC::MCA_MISC0_UMC)

Log miscellaneous information associated with errors.

[_inst\[UMCWPHY\[8,6,4,2,10,0\]UMC\]_n\[10,8,6,4,2,0\]_umc0_aliasMSRLEGACY; MSR0000_0457](#)[_inst\[UMCWPHY\[9,7,5,3,1,11\]UMC\]_n\[11,9,7,5,3,1\]_umc0_aliasMSRLEGACY; MSR0000_045B](#)[_inst\[UMCWPHY\[8,6,4,2,10,0\]UMC\]_n\[10,8,6,4,2,0\]_umc0_aliasMSR; MSRC000_2153](#)[_inst\[UMCWPHY\[9,7,5,3,1,11\]UMC\]_n\[11,9,7,5,3,1\]_umc0_aliasMSR; MSRC000_2163](#)

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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::UMC::MCA_MISC0_UMC[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	LvtOffset. 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::UMC::MCA_MISC0_UMC[Locked]) ? Read-write : Read-only.
51	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msrr::HWCR[McStatusWrEn] !MCA::UMC::MCA_MISC0_UMC[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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::UMC::MCA_MISC0_UMC[Locked]) ? Read-write : Read-only.
48	Ovrflw. 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::UMC::MCA_MISC0_UMC[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	ErrCnt. 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::UMC::MCA_MISC0_UMC[Locked]) ? Read-write : Read-only.
31:24	BlkPtr. Read-write. Reset: 01h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

MSRC000_21[5...6]4 [UMC Machine Check Configuration] (MCA::UMC::MCA_CONFIG_UMC)

Reset: 0000_0000_0000_0125h.

Controls configuration of the associated machine check bank.

_inst[UMCWPHY[8,6,4,2,10,0]UMC]_n[10,8,6,4,2,0]_umc0_aliasMSR; MSRC000_2154

_inst[UMCWPHY[9,7,5,3,1,11]UMC]_n[11,9,7,5,3,1]_umc0_aliasMSR; MSRC000_2164

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	LogDeferredInMcaStat. 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	McaXEnable. 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	IntPresent. 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	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,1. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::UMC::MCA_CONFIG_UMC[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. 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	DeferredErrorLoggingSupported. 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	McaX. 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_21[5...6]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.

_inst[UMCWPHY[8,6,4,2,10,0]UMC]_n[10,8,6,4,2,0]_umc0_aliasMSR; MSRC000_2155

_inst[UMCWPHY[9,7,5,3,1,11]UMC]_n[11,9,7,5,3,1]_umc0_aliasMSR; MSRC000_2165

Bits	Description
63:48	McaType . Read-only. Reset: 0000h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi . 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	HardwareID . Read-only. Reset: 096h. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _instUMCWPHY0UMC_n0_umc0_aliasMSR: 0005_0F00h
	Init: _instUMCWPHY10UMC_n10_umc0_aliasMSR: 00A5_0F00h
	Init: _instUMCWPHY11UMC_n11_umc0_aliasMSR: 00B5_0F00h
	Init: _instUMCWPHY1UMC_n1_umc0_aliasMSR: 0015_0F00h
	Init: _instUMCWPHY2UMC_n2_umc0_aliasMSR: 0025_0F00h
	Init: _instUMCWPHY3UMC_n3_umc0_aliasMSR: 0035_0F00h
	Init: _instUMCWPHY4UMC_n4_umc0_aliasMSR: 0045_0F00h
	Init: _instUMCWPHY5UMC_n5_umc0_aliasMSR: 0055_0F00h
	Init: _instUMCWPHY6UMC_n6_umc0_aliasMSR: 0065_0F00h
	Init: _instUMCWPHY7UMC_n7_umc0_aliasMSR: 0075_0F00h
	Init: _instUMCWPHY8UMC_n8_umc0_aliasMSR: 0085_0F00h
	Init: _instUMCWPHY9UMC_n9_umc0_aliasMSR: 0095_0F00h

MSRC000_21[5...6]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](#)

_inst[UMCWPHY[8,6,4,2,10,0]UMC]_n[10,8,6,4,2,0]_umc0_aliasMSR; MSRC000_2156

_inst[UMCWPHY[9,7,5,3,1,11]UMC]_n[11,9,7,5,3,1]_umc0_aliasMSR; MSRC000_2166

Bits	Description
63:32	Syndrome . 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	ErrorPriority . 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	Length . Read-write, Volatile . Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::UMC::MCA_SYND_UMC[Syndrome]. Length values greater than 32 (decimal) are interpreted as equal to 32 (decimal). 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	ErrorInformation . Read-write, Volatile . Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 81 [MCA_SYND_UMC].

Table 81: MCA_SYND_UMC

Error Type	Bits	Description
DramEccErr	[17:16]	Reserved
	[15]	Software-Managed Bad Symbol ID Error
	[14]	Reserved

	[13:8] [7:4] [3] [2:0]	Symbol. Only contains valid information on a corrected error. Cid. Specifies the rank multiply ID for supported DIMMs. Sub channel 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	[17:12] [11:9] [11:8] [7:4] [3] [2:0]	Reserved DevID of x8 Dram with highest Error count DevID of x4 Dram with highest Error Count CID (Specifies the rank multiply ID for supported DIMMs.) Subchannel (0=A, 1=B) Chip Select The address that is reported for ECS errors is the address of the DDR5 DRAM. This information refers to a specific location in DDR5 device that needs to be repaired (not the system address of the event). Address[63:0] = {40'd0,BG[2:0],BA[1:0]Row[17:0]} (This information is derived from MR18, MR17, MR16.)
EcsErr	[17:12] [11:9] [11:8] [7:4] [3] [2:0]	Reserved DevID of x8 Dram with highest Error count DevID of x4 Dram with highest Error Count CID (Specifies the rank multiply ID for supported DIMMs.) Subchannel (0=A, 1=B) Chip Select
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
RfmSramEccErr	[17:13] [12:0]	Reserved Reserved

MSRC000_21[5...6]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](#)[_inst\[UMCWPHY\[8,6,4,2,10,0\]UMC\]_n\[10,8,6,4,2,0\]_umc0_aliasMSR; MSRC000_215E](#)[_inst\[UMCWPHY\[9,7,5,3,1,11\]UMC\]_n\[11,9,7,5,3,1\]_umc0_aliasMSR; MSRC000_216E](#)

Bits	Description
63:0	Syndrone. 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_21[5...6]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](#)[_inst\[UMCWPHY\[8,6,4,2,10,0\]UMC\]_n\[10,8,6,4,2,0\]_umc0_aliasMSR; MSRC000_215F](#)[_inst\[UMCWPHY\[9,7,5,3,1,11\]UMC\]_n\[11,9,7,5,3,1\]_umc0_aliasMSR; MSRC000_216F](#)

Bits	Description
63:0	Syndrone. 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_21[5...6]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.

_inst[UMCWPHY[8,6,4,2,10,0]UMC]_n[10,8,6,4,2,0]_umc0_aliasMSR; MSRC000_2158_inst[UMCWPHY[9,7,5,3,1,11]UMC]_n[11,9,7,5,3,1]_umc0_aliasMSR; MSRC000_2168

Bits	Description
63	Val. Read-write, Volatile . Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	Overflow. 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 overwrite priorities.)
61:59	RESERV4. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	AddrV. Read-write, Volatile . Reset: Cold,0. 1=MCA::UMC::MCA_DEADDR_UMC contains address information associated with the error.
57:54	RESERV3. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	SyndV. 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	RESERV2. Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	Deferred. 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	RESERV1. Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	AddrLsb. 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	RESERV0. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	ErrorCodeExt. 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	ErrorCode. Read-write, Volatile . Reset: Cold,0000h. Error code for this error.

MSRC000_21[5...6]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].

_inst[UMCWPHY[8,6,4,2,10,0]UMC]_n[10,8,6,4,2,0]_umc0_aliasMSR; MSRC000_2159_inst[UMCWPHY[9,7,5,3,1,11]UMC]_n[11,9,7,5,3,1]_umc0_aliasMSR; MSRC000_2169

Bits	Description
63:0	ErrorAddr. 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_21[5...6]A [UMC Machine Check Miscellaneous 1] (MCA::UMC::MCA_MISC1_UMC)

Log miscellaneous information associated with errors, as defined by each error type.

_inst[UMCWPHY[8,6,4,2,10,0]UMC]_n[10,8,6,4,2,0]_umc0_aliasMSR; MSRC000_215A

_inst[UMCWPHY[9,7,5,3,1,11]UMC]_n[11,9,7,5,3,1]_umc0_aliasMSR; MSRC000_216A

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn] !MCA::UMC::MCA_MISC1_UMC[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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	Ovrflw. 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	ErrCnt. 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	BlkPtr. Read-write. Reset: 01h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

MSRC001_041[5...6] [UMC Machine Check Control Mask] (MCA::UMC::MCA_CTL_MASK_UMC)

Read-write. Reset: 0000_0000_0000_0000h.

Inhibit detection of an error source.

`_inst[UMCWPHY[8,6,4,2,10,0]UMC]_n[10,8,6,4,2,0]_umc0_aliasMSR; MSRC001_0415``_inst[UMCWPHY[9,7,5,3,1,11]UMC]_n[11,9,7,5,3,1]_umc0_aliasMSR; MSRC001_0416`

Bits	Description
63:17	Reserved.
16	RfmSramEccErr. Read-write. Reset: 0. RFM SRAM ECC error. An ECC error occurred on a RFM SRAM in the processor.
15:12	Reserved.
11	RdCrcErr. Read-write. Reset: 0. Read CRC error. CRC error occurred on a DRAM read from any subchannel
10	ThrttlErr. Read-write. Reset: 0. Indicates that UMC is throttling
9	EcsErr. Read-write. Reset: 0. ECS Error. Indicates that a device exceeded the ECS Error Threshold Count.
8	EcsRowErr. Read-write. Reset: 0. ECS Row Error. Indicates that a single device row exceeded four code word errors.
7	AesSramEccErr. Read-write. Reset: 0. AES SRAM ECC error. An ECC error occurred on a AES SRAM in the processor.
6	DcqSramEccErr. Read-write. Reset: 0. DCQ SRAM ECC error. An ECC error occurred on a DCQ SRAM in the processor.
5	WriteDataCrcErr. Read-write. Reset: 0. Write data CRC error. A write data CRC error occurred on the DRAM data bus.
4	AddressCommandParityErr. Read-write. Reset: 0. Address/Command parity error. A parity error occurred on the DRAM address/command bus.
3	ApbErr. Read-write. Reset: 0. Advanced peripheral bus error. An error occurred on the advanced peripheral bus.
2	SdpParityErr. Read-write. Reset: 0. SDP parity error. A parity error was detected on write data from the data fabric in the processor.
1	WriteDataPoisonErr. 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	DramEccErr. Read-write. Reset: 0. DRAM ECC error. An ECC error occurred on a DRAM read.

3.2.5.11 PSP

MSR0000_0478...MSRC000_21E0 [PSP Machine Check Control] (MCA::PSP::MCA_CTL_PSP)

0=Disables error reporting for the corresponding error. 1=Enables error reporting via machine check exception for the corresponding error. The MCA::PSP::MCA_CTL_PSP register must be enabled by the corresponding enable bit in Core::X86::Msrr::MCG_CTL. Does not affect error detection, correction, or logging.

_instMPASMPASP_n0_aliasMSRLEGACY; MSR0000_0478

_instMPASMPASP_n0_aliasMSR; MSRC000_21E0

Bits	Description
63	WAFL. Read-only. A corrected CRC or uncorrected error occurred on the WAFL link
62:24	Reserved.
23	MpASPSibWdtError. Read-write. Reset: 0. SIB WDT error.
22	MpASPA5Hang. Read-write. Reset: 0. mpASP A5 Hang.
21	MpASPSecEMCError. Read-write. Reset: 0. mpASP SEC EMC Error.
20	MpASPSibSramError. Read-write. Reset: 0. SIB SRAM parity error.
19	MpASPPcruFuseSramError. Read-write. Reset: 0. PCRU FUSE SRAM ECC or parity error.
18	MpASPFuseSramError. Read-write. Reset: 0. FUSE IP SRAM ECC or parity error.
17	MpASPSHubIfRdBufError. Read-write. Reset: 0. System Hub Read Buffer ECC or parity error.
16	Mp0TlbBank1Error. Read-write. Reset: 0. TLB Bank 1 parity error.
15	MpASPTlbBank0Error. Read-write. Reset: 0. TLB Bank 0 parity error.
14	MpASPDDirtyRamError. Read-write. Reset: 0. Dirty Data Ram parity error.
13	MpASPDTagBank3Error. Read-write. Reset: 0. Data Tag Bank 3 parity error.
12	MpASPDTagBank2Error. Read-write. Reset: 0. Data Tag Bank 2 parity error.
11	MpASPDTagBank1Error. Read-write. Reset: 0. Data Tag Bank 1 parity error.
10	MpASPDTagBank0Error. Read-write. Reset: 0. Data Tag Bank 0 parity error.
9	MpASPDDDataBank3Error. Read-write. Reset: 0. Data Cache Bank 3 ECC or parity error.
8	MpASPDDDataBank2Error. Read-write. Reset: 0. Data Cache Bank 2 ECC or parity error.
7	MpASPDDDataBank1Error. Read-write. Reset: 0. Data Cache Bank 1 ECC or parity error.
6	MpASPDDDataBank0Error. Read-write. Reset: 0. Data Cache Bank 0 ECC or parity error.
5	MpASPITagRam1Error. Read-write. Reset: 0. Instruction Tag Ram 1 parity error.
4	MpASPITagRam0Error. Read-write. Reset: 0. Instruction Tag Ram 0 parity error.
3	MpASPIDataBank1Error. Read-write. Reset: 0. Instruction Cache Bank 1 ECC or parity error.
2	MpASPIDataBank0Error. Read-write. Reset: 0. Instruction Cache Bank 0 ECC or parity error.
1	MpASPLowSramError. Read-write. Reset: 0. Low SRAM ECC or parity error.
0	MpASPHighSramError. Read-write. Reset: 0. High SRAM ECC or parity error.

MSR0000_0479...MSRC000_21E1 [PSP Machine Check Status] (MCA::PSP::MCA_STATUS_PSP)

Reset: Cold,0000_0000_0000_0000h.

Logs information associated with errors.

_instMPASMPASP_n0_aliasMSRLEGACY; MSR0000_0479

_instMPASMPASP_n0_aliasMSR; MSRC000_21E1

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::PSP::MCA_CTL_PSP. This bit is a copy of bit in MCA::PSP::MCA_CTL_PSP for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	MiscV. Reset: Cold,0. 1=Valid thresholding in MCA::PSP::MCA_MISC0_PSP. 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	AddrV. Reset: Cold,0. 1=MCA::PSP::MCA_ADDR_PSP contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	PCC. 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	ErrCoreIdVal. 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	TCC. 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::PSP::MCA_STATUS_PSP[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	RESERV54. 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	SyndV. Reset: Cold,0. 1=This error logged information in MCA::PSP::MCA_SYND_PSP. If MCA::PSP::MCA_SYND_PSP[ErrorPriority] is the same as the priority of the error in MCA::PSP::MCA_STATUS_PSP, then the information in MCA::PSP::MCA_SYND_PSP is associated with the error in MCA::PSP::MCA_STATUS_PSP. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::PSP::MCA_ADDR_PSP[ErrorAddr]. A value of 0 indicates that MCA::PSP::MCA_ADDR_PSP[63:0] contains a valid byte address. A value of 6 indicates that MCA::PSP::MCA_ADDR_PSP[63:6] contains a valid cache line address and that MCA::PSP::MCA_ADDR_PSP[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::PSP::MCA_ADDR_PSP[63:12] contain a valid 4KB memory page and that MCA::PSP::MCA_ADDR_PSP[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	RESERV22. 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	ErrorCodeExt. 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::PSP::MCA_CTL_PSP 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	ErrorCode. 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 82: MCA_STATUS_PSP

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
MpASPHigh SramError	0x0	0/1	0/1	0/1	0	0	1
MpASPLow SramError	0x1	0/1	0/1	0/1	0	0	1
MpASPIDat aBank0Error	0x2	0/1	0/1	0/1	0	0	1
MpASPIDat aBank1Error	0x3	0/1	0/1	0/1	0	0	1
MpASPITag Ram0Error	0x4	1	1	1	0	0	1
MpASPITag Ram1Error	0x5	1	1	1	0	0	1
MpASPDDa	0x6	0/1	0/1	0/1	0	0	1

taBank0Error							
MpASPDDa taBank1Error	0x7	0/1	0/1	0/1	0	0	1
MpASPDDa taBank2Error	0x8	0/1	0/1	0/1	0	0	1
MpASPDDa taBank3Error	0x9	0/1	0/1	0/1	0	0	1
MpASPDTagBank0Error	0xa	1	1	1	0	0	1
MpASPDTagBank1Error	0xb	1	1	1	0	0	1
MpASPDTagBank2Error	0xc	1	1	1	0	0	1
MpASPDTagBank3Error	0xd	1	1	1	0	0	1
MpASPDDir tyRamError	0xe	1	1	1	0	0	1
MpASPTlbBank0Error	0xf	1	1	1	0	0	1
Mp0TlbBank1Error	0x10	1	1	1	0	0	1
MpASPSHubIfRdBufError	0x11	1	1	1	0	0	1
MpASPFuse SramError	0x12	0/1	0/1	0/1	0	0	1
MpASPPcru FuseSramError	0x13	0/1	0/1	0/1	0	0	1
MpASPSibS ramError	0x14	1	1	1	0	0	1
MpASPSecE MCErr	0x15	1	1	1	0	0	1
MpASPA5H ang	0x16	1	1	1	0	0	1
MpASPSib WdtError	0x17	1	1	1	0	0	1
Reserved	0x3a	1	1	1	0	0	1
Reserved	0x3b	1	1	1	0	0	1
Reserved	0x3c	1	1	1	0	0	1
Reserved	0x3d	1	1	1	0	0	1
Reserved	0x3e	1	1	1	0	0	1
WAFL	0x3f	0/1	0/1	0/1	0	0	1

MSR0000_047A...MSRC000_21E2 [PSP Machine Check Address] (MCA::PSP::MCA_ADDR_PSP)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

MCA::PSP::MCA_ADDR_PSP stores an address and other information associated with the error in MCA::PSP::MCA_STATUS_PSP. The register is only meaningful if MCA::PSP::MCA_STATUS_PSP[Val]=1 and MCA::PSP::MCA_STATUS_PSP[AddrV]=1.

_instMPASMPASP_n0_aliasMSRLEGACY; MSR0000_047A

_instMPASMPASP_n0_aliasMSR; MSRC000_21E2

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::PSP::MCA_STATUS_PSP. For physical addresses, the most significant bit is given by Core::X86::Cpuid::LongModeInfo[PhysAddrSize].

Table 83: MCA_ADDR_PSP

Error Type	Bits	Description
MpASPHighSramError	[55:0]	Reserved
MpASPLowSramError	[55:0]	Reserved
MpASPIDataBank0Error	[55:0]	Reserved
MpASPIDataBank1Error	[55:0]	Reserved
MpASPIDTagRam0Error	[55:0]	Reserved
MpASPIDTagRam1Error	[55:0]	Reserved
MpASPDDDataBank0Error	[55:0]	Reserved
MpASPDDDataBank1Error	[55:0]	Reserved
MpASPDDDataBank2Error	[55:0]	Reserved
MpASPDDDataBank3Error	[55:0]	Reserved
MpASPDDTagBank0Error	[55:0]	Reserved
MpASPDDTagBank1Error	[55:0]	Reserved
MpASPDDTagBank2Error	[55:0]	Reserved
MpASPDDTagBank3Error	[55:0]	Reserved
MpASPDDirtyRamError	[55:0]	Reserved
MpASPTlbBank0Error	[55:0]	Reserved
Mp0TlbBank1Error	[55:0]	Reserved
MpASPSHubIfRdBufError	[55:0]	Reserved
MpASPFuseSramError	[55:0]	Reserved
MpASPPcruFuseSramError	[55:0]	Reserved
MpASPSibSramError	[55:0]	Reserved
MpASPSecEMCError	[55:0]	Reserved
MpASPA5Hang	[55:0]	Reserved
MpASPSibWdtError	[55:0]	Reserved
Reserved	[55:0]	Reserved
Reserved	[55:0]	Reserved
Reserved	[55:0]	Reserved
Reserved	[55:0]	Reserved
Reserved	[55:0]	Reserved
WAFL	[55:0]	Reserved

MSR0000_047B...MSRC000_21E3 [PSP Machine Check Miscellaneous 0] (MCA::PSP::MCA_MISC0_PSP)

Log miscellaneous information associated with errors.

_instMPASMPASP_n0_aliasMSRLEGACY; MSR0000_047B

_instMPASMPASP_n0_aliasMSR; MSRC000_21E3

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr:: HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr:: HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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::PSP::MCA_MISC0_PSP[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	LvtOffset. 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::PSP::MCA_MISC0_PSP[Locked]) ? Read-write : Read-only.
51	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr:: HWCR[McStatusWrEn] !MCA::PSP::MCA_MISC0_PSP[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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::PSP::MCA_MISC0_PSP[Locked]) ? Read-write : Read-only.
48	Ovrflw. 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::PSP::MCA_MISC0_PSP[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	ErrCnt. 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::PSP::MCA_MISC0_PSP[Locked]) ? Read-write : Read-only.
31:24	BlkPtr. Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

MSRC000_21E4 [PSP Machine Check Configuration] (MCA::PSP::MCA_CONFIG_PSP)

Reset: 0000_0000_0000_0121h.

Controls configuration of the associated machine check bank.

_instMPASPMASP_n0_aliasMSR; MSRC000_21E4

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	McaXEnable. 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	IntPresent. Read-only, Volatile . Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::PSP::MCA_CONFIG_PSP[IntEn].
9	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::PSP::MCA_CONFIG_PSP[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. Read-only. Reset: 1. 1=MCA::PSP::MCA_CONFIG_PSP[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::PSP::MCA_CONFIG_PSP[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	DeferredErrorLoggingSupported. 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	McaX. Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::PSP::MCA_MISC0_PSP[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::PSP::MCA_STATUS_PSP[TCC] is present.

MSRC000_21E5 [PSP IP Identification] (MCA::PSP::MCA_IPID_PSP)

Reset: 0001_00FF_0000_0000h.

The MCA::PSP::MCA_IPID_PSP register is used by software to determine what IP type and revision is associated with the MCA bank.

_instMPASPMASP_n0_aliasMSR; MSRC000_21E5

Bits	Description
63:48	McaType. Read-only. Reset: 0001h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi. 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	HardwareID. Read-only. Reset: 0FFh. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId. Read-write. Reset: 0000_0000h. Init: 0381_2000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.

MSRC000_21E6 [PSP Machine Check Syndrome] (MCA::PSP::MCA_SYND_PSP)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::PSP::MCA_STATUS_PSP [Thread 0](#)

_instMPASPMPPASP_n0_aliasMSR; MSRC000_21E6

Bits	Description
63:27	Reserved.
26:24	ErrorPriority. Read-write, Volatile . Reset: Cold, 0h. Encodes the priority of the error logged in MCA::PSP::MCA_SYND_PSP. 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	Length. 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	ErrorInformation. Read-write, Volatile . Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 84 [MCA_SYND_PSP].

Table 84: MCA_SYND_PSP

Error Type	Bits	Description
MpASPHighSramError	[17:0]	Reserved
MpASPLowSramError	[17:0]	Reserved
MpASPIDataBank0Error	[17:9] [8:0]	Reserved Reserved
MpASPIDataBank1Error	[17:9] [8:0]	Reserved Reserved
MpASPIDTagRam0Error	[17:7] [6:0]	Reserved Reserved
MpASPIDTagRam1Error	[17:7] [6:0]	Reserved Reserved
MpASPDDataBank0Error	[17:9] [8:0]	Reserved Reserved
MpASPDDataBank1Error	[17:9] [8:0]	Reserved Reserved
MpASPDDataBank2Error	[17:9] [8:0]	Reserved Reserved
MpASPDDataBank3Error	[17:9] [8:0]	Reserved Reserved
MpASPDDTagBank0Error	[17:6] [5:0]	Reserved Reserved
MpASPDDTagBank1Error	[17:6] [5:0]	Reserved Reserved
MpASPDDTagBank2Error	[17:6] [5:0]	Reserved Reserved
MpASPDDTagBank3Error	[17:6] [5:0]	Reserved Reserved
MpASPDDirtyRamError	[17:6] [5:0]	Reserved Reserved
MpASPTlbBank0Error	[17:6] [5:0]	Reserved Reserved
Mp0TlbBank1Error	[17:6] [5:0]	Reserved Reserved
MpASPSHubIfRdBufError	[17:6]	Reserved

	[5:0]	Reserved
MpASPFuseSramError	[17:0]	Reserved
MpASPPcruFuseSramError	[17:0]	Reserved
MpASPSibSramError	[17:0]	Reserved
MpASPSecEMCError	[17:0]	Reserved
MpASPA5Hang	[17:0]	Reserved
MpASPSibWdtError	[17:0]	Reserved
Reserved	[17:0]	Reserved
Reserved	[17:0]	Reserved
Reserved	[17:0]	Reserved
Reserved	[17:0]	Reserved
Reserved	[32:0]	Reserved
WAFL	[17:0]	Reserved

MSRC001_041E [PSP Machine Check Control Mask] (MCA::PSP::MCA_CTL_MASK_PSP)

Inhibit detection of an error source.

_instMPASPMASP_n0_aliasMSR; MSRC001_041E

Bits	Description
63	WAFL. Read-only. A corrected CRC or uncorrected error occurred on the WAFL link
62:24	Reserved.
23	MpASPSibWdtError. Read-write. Reset: 0. SIB WDT error.
22	MpASPA5Hang. Read-write. Reset: 0. mpASP A5 Hang.
21	MpASPSecEMCError. Read-write. Reset: 0. mpASP SEC EMC Error.
20	MpASPSibSramError. Read-write. Reset: 0. SIB SRAM parity error.
19	MpASPPcruFuseSramError. Read-write. Reset: 0. PCRU FUSE SRAM ECC or parity error.
18	MpASPFuseSramError. Read-write. Reset: 0. FUSE IP SRAM ECC or parity error.
17	MpASPSHubIfRdBufError. Read-write. Reset: 0. System Hub Read Buffer ECC or parity error.
16	Mp0TlbBank1Error. Read-write. Reset: 0. TLB Bank 1 parity error.
15	MpASPTlbBank0Error. Read-write. Reset: 0. TLB Bank 0 parity error.
14	MpASPDDirtyRamError. Read-write. Reset: 0. Dirty Data Ram parity error.
13	MpASPDTagBank3Error. Read-write. Reset: 0. Data Tag Bank 3 parity error.
12	MpASPDTagBank2Error. Read-write. Reset: 0. Data Tag Bank 2 parity error.
11	MpASPDTagBank1Error. Read-write. Reset: 0. Data Tag Bank 1 parity error.
10	MpASPDTagBank0Error. Read-write. Reset: 0. Data Tag Bank 0 parity error.
9	MpASPDDDataBank3Error. Read-write. Reset: 0. Data Cache Bank 3 ECC or parity error.
8	MpASPDDDataBank2Error. Read-write. Reset: 0. Data Cache Bank 2 ECC or parity error.
7	MpASPDDDataBank1Error. Read-write. Reset: 0. Data Cache Bank 1 ECC or parity error.
6	MpASPDDDataBank0Error. Read-write. Reset: 0. Data Cache Bank 0 ECC or parity error.
5	MpASPIITagRam1Error. Read-write. Reset: 0. Instruction Tag Ram 1 parity error.
4	MpASPIITagRam0Error. Read-write. Reset: 0. Instruction Tag Ram 0 parity error.
3	MpASPIDataBank1Error. Read-write. Reset: 0. Instruction Cache Bank 1 ECC or parity error.
2	MpASPIDataBank0Error. Read-write. Reset: 0. Instruction Cache Bank 0 ECC or parity error.
1	MpASPLowSramError. Read-write. Reset: 0. Low SRAM ECC or parity error.
0	MpASPHighSramError. Read-write. Reset: 0. High SRAM ECC or parity error.

MSRC000_21EE [PSP Machine Check Syndrome Extended] (MCA::PSP::MCA_SYND1_PSP)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::PSP::MCA_STATUS_PSP [Thread 0](#)[_instMPASMPASP_n0_aliasMSR; MSRC000_21EE](#)

Bits	Description
63:0	Syndrome. Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::PSP::MCA_SYND1_PSP register stores information associated with the error in MCA::PSP::MCA_STATUS_PSP or MCA_DESTAT. The register is meaningful if MCA::PSP::MCA_STATUS_PSP[SyndV]=1. When MCA::PSP::MCA_CONFIG_PSP[McaFruTextInMca]=1, MCA::PSP::MCA_SYND1_PSP stores ASCII FruText associated with the error.

MSRC000_21EF [PSP Machine Check Syndrome Extended] (MCA::PSP::MCA_SYND2_PSP)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::PSP::MCA_STATUS_PSP [Thread 0](#)[_instMPASMPASP_n0_aliasMSR; MSRC000_21EF](#)

Bits	Description
63:0	Syndrome. Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::PSP::MCA_SYND2_PSP register stores information associated with the error in MCA::PSP::MCA_STATUS_PSP or MCA_DESTAT. The register is meaningful if MCA::PSP::MCA_STATUS_PSP[SyndV]=1. When MCA::PSP::MCA_CONFIG_PSP[McaFruTextInMca]=1, MCA::PSP::MCA_SYND2_PSP stores ASCII FruText associated with the error.

3.2.5.12 SMU**MSR0000_0474...MSRC000_21D0 [SMU Machine Check Control] (MCA::SMU::MCA_CTL_SMU_MP1)**

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::SMU::MCA_CTL_SMU_MP1 register must be enabled by the corresponding enable bit in Core::X86::Msr::[MCG_CTL](#). Does not affect error detection, correction, or logging.

[_instMP1MP1_n0_aliasMSRLEGACY; MSR0000_0474](#)[_instMP1MP1_n0_aliasMSR; MSRC000_21D0](#)

Bits	Description
63:11	Reserved.
10	Mp1SHubIfRdBufError. Read-write. Reset: 0. System Hub Read Buffer ECC or parity error.
9	Mp1ITagBError. Read-write. Reset: 0. Instruction Tag Cache Bank B ECC or parity error.
8	Mp1ITagAError. Read-write. Reset: 0. Instruction Tag Cache Bank A ECC or parity error.
7	Mp1ICacheBError. Read-write. Reset: 0. Instruction Cache Bank B ECC or parity error.
6	Mp1ICacheAError. Read-write. Reset: 0. Instruction Cache Bank A ECC or parity error.
5	Mp1DTagBError. Read-write. Reset: 0. Data Tag Cache Bank B ECC or parity error.
4	Mp1DTagAError. Read-write. Reset: 0. Data Tag Cache Bank A ECC or parity error.
3	Mp1DCacheBError. Read-write. Reset: 0. Data Cache Bank B ECC or parity error.
2	Mp1DCacheAError. Read-write. Reset: 0. Data Cache Bank A ECC or parity error.
1	Mp1LowSramError. Read-write. Reset: 0. Low SRAM ECC or parity error.
0	Mp1HighSramError. Read-write. Reset: 0. High SRAM ECC or parity error.

MSR0000_0474...MSRC000_21D0 [SMU Machine Check Control] (MCA::SMU::MCA_CTL_SMU_MPIO)

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::SMU::MCA_CTL_SMU_MPIO register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG_CTL. Does not affect error detection, correction, or logging.

_instMPIOMPIO_n1_aliasMSRLEGACY; MSR0000_0474

_instMPIOMPIO_n1_aliasMSR; MSRC000_21D0

Bits	Description
63:11	Reserved.
10	MpIOSHubIfRdBufError. Read-write. Reset: 0. System Hub Read Buffer ECC or parity error.
9	MpIOITagBError. Read-write. Reset: 0. Instruction Tag Cache Bank B ECC or parity error.
8	MpIOITagAError. Read-write. Reset: 0. Instruction Tag Cache Bank A ECC or parity error.
7	MpIOICacheBError. Read-write. Reset: 0. Instruction Cache Bank B ECC or parity error.
6	MpIOICacheAError. Read-write. Reset: 0. Instruction Cache Bank A ECC or parity error.
5	MpIODTagBError. Read-write. Reset: 0. Data Tag Cache Bank B ECC or parity error.
4	MpIODTagAError. Read-write. Reset: 0. Data Tag Cache Bank A ECC or parity error.
3	MpIODCacheBError. Read-write. Reset: 0. Data Cache Bank B ECC or parity error.
2	MpIODCacheAError. Read-write. Reset: 0. Data Cache Bank A ECC or parity error.
1	MpIOLowSramError. Read-write. Reset: 0. Low SRAM ECC or parity error.
0	MpIOHighSramError. Read-write. Reset: 0. High SRAM ECC or parity error.

MSR0000_0474...MSRC000_21D0 [SMU Machine Check Control] (MCA::SMU::MCA_CTL_SMU_MPRAS)

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::SMU::MCA_CTL_SMU_MPRAS register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG_CTL. Does not affect error detection, correction, or logging.

_instMPRASMPRAS_n2_aliasMSRLEGACY; MSR0000_0474

_instMPRASMPRAS_n2_aliasMSR; MSRC000_21D0

Bits	Description
63:11	Reserved.
10	MpRASSHubIfRdBufError. Read-write. Reset: 0. System Hub Read Buffer ECC or parity error.
9	MpRASITagBError. Read-write. Reset: 0. Instruction Tag Cache Bank B ECC or parity error.
8	MpRASITagAError. Read-write. Reset: 0. Instruction Tag Cache Bank A ECC or parity error.
7	MpRASICacheBError. Read-write. Reset: 0. Instruction Cache Bank B ECC or parity error.
6	MpRASICacheAError. Read-write. Reset: 0. Instruction Cache Bank A ECC or parity error.
5	MpRASDTagBError. Read-write. Reset: 0. Data Tag Cache Bank B ECC or parity error.
4	MpRASDTagAError. Read-write. Reset: 0. Data Tag Cache Bank A ECC or parity error.
3	MpRASDCacheBError. Read-write. Reset: 0. Data Cache Bank B ECC or parity error.
2	MpRASDCacheAError. Read-write. Reset: 0. Data Cache Bank A ECC or parity error.
1	MpRASLowSramError. Read-write. Reset: 0. Low SRAM ECC or parity error.
0	MpRASHighSramError. Read-write. Reset: 0. High SRAM ECC or parity error.

MSR0000_0475...MSRC000_21D1 [SMU Machine Check Status] (MCA::SMU::MCA_STATUS_SMU)

Reset: Cold,0000_0000_0000_0000h.

Logs information associated with errors.

_inst[MPRASMPPRAS,MPIOMPIO,MP1MP1]_n[2:0]_aliasMSRLEGACY; MSR0000_0475

_inst[MPRASMPPRAS,MPIOMPIO,MP1MP1]_n[2:0]_aliasMSR; MSRC000_21D1

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::SMU::MCA_CTL_SMU_MP1 or MCA::SMU::MCA_CTL_SMU_MPIO or MCA::SMU::MCA_CTL_SMU_MPPRAS. This bit is a copy of bit in MCA::SMU::MCA_CTL_SMU_MP1 or MCA::SMU::MCA_CTL_SMU_MPIO or MCA::SMU::MCA_CTL_SMU_MPPRAS for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	MiscV. Reset: Cold,0. 1=Valid thresholding in MCA::SMU::MCA_MISC0_SMU. 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	AddrV. Reset: Cold,0. 1=MCA::SMU::MCA_ADDR_SMU contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	PCC. 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	ErrCoreIdVal. 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	TCC. 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::SMU::MCA_STATUS_SMU[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	RESERV54. 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	SyndV. Reset: Cold,0. 1=This error logged information in MCA::SMU::MCA_SYND_SMU. If MCA::SMU::MCA_SYND_SMU[ErrorPriority] is the same as the priority of the error in MCA::SMU::MCA_STATUS_SMU, then the information in MCA::SMU::MCA_SYND_SMU is associated with the error in MCA::SMU::MCA_STATUS_SMU. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::SMU::MCA_ADDR_SMU[ErrorAddr]. A value of 0 indicates that MCA::SMU::MCA_ADDR_SMU[63:0] contains a valid byte address. A value of 6 indicates that MCA::SMU::MCA_ADDR_SMU[63:6] contains a valid cache line address and that MCA::SMU::MCA_ADDR_SMU[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::SMU::MCA_ADDR_SMU[63:12] contain a valid 4KB memory page and that MCA::SMU::MCA_ADDR_SMU[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	RESERV22. 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	ErrorCodeExt. 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::SMU::MCA_CTL_SMU_MP1 or MCA::SMU::MCA_CTL_SMU_MPIO or MCA::SMU::MCA_CTL_SMU_MPRAS 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	ErrorCode. 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 85: MCA_STATUS_SMU

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
MpIOHighSramError	0x0	0/1	0/1	0/1	0	0	1
MpIOLowSramError	0x1	0/1	0/1	0/1	0	0	1
MpIODCacheError	0x2	0/1	0/1	0/1	0	0	1
MpIODCacheError	0x3	0/1	0/1	0/1	0	0	1

MpIODTag AError	0x4	0/1	0/1	0/1	0	0	1
MpIODTagB Error	0x5	0/1	0/1	0/1	0	0	1
MpIOICache AError	0x6	0/1	0/1	0/1	0	0	1
MpIOICache BError	0x7	0/1	0/1	0/1	0	0	1
MpIOITagA Error	0x8	0/1	0/1	0/1	0	0	1
MpIOITagB Error	0x9	0/1	0/1	0/1	0	0	1
MpIOSHubI fRdBufError	0xa	0/1	0/1	0/1	0	0	1

MSR0000_0476...MSRC000_21D2 [SMU Machine Check Address] (MCA::SMU::MCA_ADDR_SMU)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

MCA::SMU::MCA_ADDR_SMU stores an address and other information associated with the error in MCA::SMU::MCA_STATUS_SMU. The register is only meaningful if MCA::SMU::MCA_STATUS_SMU[Val]=1 and MCA::SMU::MCA_STATUS_SMU[AddrV]=1.

_inst[MPRASMPRAS, MPIOMPPIO, MP1MP1]_n[2:0]_aliasMSRLEGACY; MSR0000_0476

_inst[MPRASMPRAS, MPIOMPPIO, MP1MP1]_n[2:0]_aliasMSR; MSRC000_21D2

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::SMU::MCA_STATUS_SMU. For physical addresses, the most significant bit is given by Core::X86::Cpuid::LongModeInfo[PhysAddrSize].

Table 86: MCA_ADDR_SMU

Error Type	Bits	Description
MpIOHighSramError	[55:0]	Reserved
MpIOLowSramError	[55:0]	Reserved
MpIODCacheAError	[55:0]	Reserved
MpIODCacheBError	[55:0]	Reserved
MpIODTagAError	[55:0]	Reserved
MpIODTagBError	[55:0]	Reserved
MpIOICacheAError	[55:0]	Reserved
MpIOICacheBError	[55:0]	Reserved
MpIOITagAError	[55:0]	Reserved
MpIOITagBError	[55:0]	Reserved
MpIOSHubIfRdBufError	[55:0]	Reserved

MSR0000_0477...MSRC000_21D3 [SMU Machine Check Miscellaneous 0] (MCA::SMU::MCA_MISC0_SMU)

Log miscellaneous information associated with errors.

_inst[MPRASMPPRAS,MPIOMPIO,MP1MP1]_n[2:0]_aliasMSRLEGACY; MSR0000_0477

_inst[MPRASMPPRAS,MPIOMPIO,MP1MP1]_n[2:0]_aliasMSR; MSRC000_21D3

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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::SMU::MCA_MISC0_SMU[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	LvtOffset. 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::SMU::MCA_MISC0_SMU[Locked]) ? Read-write : Read-only.
51	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn] !MCA::SMU::MCA_MISC0_SMU[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. Reset: Cold,0h. Specifies the type of interrupt signaled when Ovrlf 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::SMU::MCA_MISC0_SMU[Locked]) ? Read-write : Read-only.
48	Ovrlf. 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::SMU::MCA_MISC0_SMU[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	ErrCnt. 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::SMU::MCA_MISC0_SMU[Locked]) ? Read-write : Read-only.
31:24	BlkPtr. Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

MSRC000_21D4 [SMU Machine Check Configuration] (MCA::SMU::MCA_CONFIG_SMU)

Reset: 0000_0000_0000_0121h.

Controls configuration of the associated machine check bank.

_inst[MPRASMPRAS,MPIOMPIO,MP1MP1]_n[2:0]_aliasMSR; MSRC000_21D4

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	McaXEnable. 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	IntPresent. Read-only, Volatile . Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::SMU::MCA_CONFIG_SMU[IntEn].
9	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::SMU::MCA_CONFIG_SMU[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. Read-only. Reset: 1. 1=MCA::SMU::MCA_CONFIG_SMU[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::SMU::MCA_CONFIG_SMU[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	DeferredErrorLoggingSupported. 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	McaX. Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::SMU::MCA_MISC0_SMU[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::SMU::MCA_STATUS_SMU[TCC] is present.

MSRC000_21D5 [SMU IP Identification] (MCA::SMU::MCA_IPID_SMU)

Reset: 0001_0001_0000_0000h.

The MCA::SMU::MCA_IPID_SMU register is used by software to determine what IP type and revision is associated with the MCA bank.

_inst[MPRASMPRAS,MPIOMPIO,MP1MP1]_n[2:0]_aliasMSR; MSRC000_21D5

Bits	Description
63:48	McaType. Read-only. Reset: 0001h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi. 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	HardwareID. Read-only. Reset: 001h. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId. Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _instMP1MP1_n0_aliasMSR: 03B3_0400h
	Init: _instMPIOMPIO_n1_aliasMSR: 0C93_0400h
	Init: _instMPRASMPRAS_n2_aliasMSR: 03E3_0400h

MSRC000_21D6 [SMU Machine Check Syndrome] (MCA::SMU::MCA_SYND_SMU)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::SMU::MCA_STATUS_SMU [Thread 0](#)

_inst[MPRASMPRAS,MPIOMPIO,MP1MP1]_n[2:0]_aliasMSR; MSRC000_21D6

Bits	Description
63:27	Reserved.
26:24	ErrorPriority. Read-write, Volatile . Reset: Cold, 0h. Encodes the priority of the error logged in MCA::SMU::MCA_SYND_SMU. 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	Length. 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	ErrorInformation. Read-write, Volatile . Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 87 [MCA_SYND_SMU].

Table 87: MCA_SYND_SMU

Error Type	Bits	Description
MpIOHighSramError	[17:15] [14:0]	Reserved Reserved
MpIOLowSramError	[17:15] [14:0]	Reserved Reserved
MpIODCacheAError	[17:8] [7:0]	Reserved Reserved
MpIODCacheBError	[17:8] [7:0]	Reserved Reserved
MpIODTagAError	[17:7] [6:0]	Reserved Reserved
MpIODTagBError	[17:7] [6:0]	Reserved Reserved
MpIOICacheAError	[17:8] [7:0]	Reserved Reserved
MpIOICacheBError	[17:8] [7:0]	Reserved Reserved
MpIOITagAError	[17:6] [5:0]	Reserved Reserved

MpIOITagBError	[17:6] [5:0]	Reserved Reserved
MpIOSHubIfRdBufError	[17:6] [5:0]	Reserved Reserved

MSRC001_041D [SMU Machine Check Control Mask] (MCA::SMU::MCA_CTL_MASK_SMU_MP1)

Read-write. Reset: 0000_0000_0000_0000h.

Inhibit detection of an error source.

_instMP1MP1_n0_aliasMSR; MSRC001_041D

Bits	Description
63:11	Reserved.
10	Mp1SHubIfRdBufError. Read-write. Reset: 0. System Hub Read Buffer ECC or parity error.
9	Mp1ITagBError. Read-write. Reset: 0. Instruction Tag Cache Bank B ECC or parity error.
8	Mp1ITagAError. Read-write. Reset: 0. Instruction Tag Cache Bank A ECC or parity error.
7	Mp1ICacheBError. Read-write. Reset: 0. Instruction Cache Bank B ECC or parity error.
6	Mp1ICacheAError. Read-write. Reset: 0. Instruction Cache Bank A ECC or parity error.
5	Mp1DTagBError. Read-write. Reset: 0. Data Tag Cache Bank B ECC or parity error.
4	Mp1DTagAError. Read-write. Reset: 0. Data Tag Cache Bank A ECC or parity error.
3	Mp1DCacheBError. Read-write. Reset: 0. Data Cache Bank B ECC or parity error.
2	Mp1DCacheAError. Read-write. Reset: 0. Data Cache Bank A ECC or parity error.
1	Mp1LowSramError. Read-write. Reset: 0. Low SRAM ECC or parity error.
0	Mp1HighSramError. Read-write. Reset: 0. High SRAM ECC or parity error.

MSRC001_041D [SMU Machine Check Control Mask] (MCA::SMU::MCA_CTL_MASK_SMU_MPIO)

Read-write. Reset: 0000_0000_0000_0000h.

Inhibit detection of an error source.

_instMPIOMPIO_n1_aliasMSR; MSRC001_041D

Bits	Description
63:11	Reserved.
10	MpIOSHubIfRdBufError. Read-write. Reset: 0. System Hub Read Buffer ECC or parity error.
9	MpIOITagBError. Read-write. Reset: 0. Instruction Tag Cache Bank B ECC or parity error.
8	MpIOITagAError. Read-write. Reset: 0. Instruction Tag Cache Bank A ECC or parity error.
7	MpIOICacheBError. Read-write. Reset: 0. Instruction Cache Bank B ECC or parity error.
6	MpIOICacheAError. Read-write. Reset: 0. Instruction Cache Bank A ECC or parity error.
5	MpIODTagBError. Read-write. Reset: 0. Data Tag Cache Bank B ECC or parity error.
4	MpIODTagAError. Read-write. Reset: 0. Data Tag Cache Bank A ECC or parity error.
3	MpIODCacheBError. Read-write. Reset: 0. Data Cache Bank B ECC or parity error.
2	MpIODCacheAError. Read-write. Reset: 0. Data Cache Bank A ECC or parity error.
1	MpIOLowSramError. Read-write. Reset: 0. Low SRAM ECC or parity error.
0	MpIOHighSramError. Read-write. Reset: 0. High SRAM ECC or parity error.

MSRC001_041D [SMU Machine Check Control Mask] (MCA::SMU::MCA_CTL_MASK_SMU_MPRAS)

Read-write. Reset: 0000_0000_0000_0000h.

Inhibit detection of an error source.

_instMPRASMPRAS_n2_aliasMSR; MSRC001_041D

Bits	Description
63:11	Reserved.
10	MpRASSHubIfRdBufError. Read-write. Reset: 0. System Hub Read Buffer ECC or parity error.
9	MpRASITagBError. Read-write. Reset: 0. Instruction Tag Cache Bank B ECC or parity error.
8	MpRASITagAError. Read-write. Reset: 0. Instruction Tag Cache Bank A ECC or parity error.
7	MpRASICacheBError. Read-write. Reset: 0. Instruction Cache Bank B ECC or parity error.
6	MpRASICacheAError. Read-write. Reset: 0. Instruction Cache Bank A ECC or parity error.
5	MpRASDTagBError. Read-write. Reset: 0. Data Tag Cache Bank B ECC or parity error.
4	MpRASDTagAError. Read-write. Reset: 0. Data Tag Cache Bank A ECC or parity error.
3	MpRASDCacheBError. Read-write. Reset: 0. Data Cache Bank B ECC or parity error.
2	MpRASDCacheAError. Read-write. Reset: 0. Data Cache Bank A ECC or parity error.
1	MpRASLowSramError. Read-write. Reset: 0. Low SRAM ECC or parity error.
0	MpRASHighSramError. Read-write. Reset: 0. High SRAM ECC or parity error.

MSRC000_21DE [SMU Machine Check Syndrome Extended] (MCA::SMU::MCA_SYND1_SMU)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::SMU::MCA_STATUS_SMU [Thread 0](#)

_inst[MPRASMPRAS, MPIOMPIO, MP1MP1]_n[2:0]_aliasMSR; MSRC000_21DE

Bits	Description
63:0	Syndrome. Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::SMU::MCA_SYND1_SMU register stores information associated with the error in MCA::SMU::MCA_STATUS_SMU or MCA_DESTAT. The register is meaningful if MCA::SMU::MCA_STATUS_SMU[SyndV]=1. When MCA::SMU::MCA_CONFIG_SMU[McaFruTextInMca]=1, MCA::SMU::MCA_SYND1_SMU stores ASCII FruText associated with the error.

MSRC000_21DF [SMU Machine Check Syndrome Extended] (MCA::SMU::MCA_SYND2_SMU)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::SMU::MCA_STATUS_SMU [Thread 0](#)

_inst[MPRASMPRAS, MPIOMPIO, MP1MP1]_n[2:0]_aliasMSR; MSRC000_21DF

Bits	Description
63:0	Syndrome. Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::SMU::MCA_SYND2_SMU register stores information associated with the error in MCA::SMU::MCA_STATUS_SMU or MCA_DESTAT. The register is meaningful if MCA::SMU::MCA_STATUS_SMU[SyndV]=1. When MCA::SMU::MCA_CONFIG_SMU[McaFruTextInMca]=1, MCA::SMU::MCA_SYND2_SMU stores ASCII FruText associated with the error.

3.2.5.13 MP5

MSR0000_043C...MSRC000_20F0 [MP5 Machine Check Control] (MCA::MP5::MCA_CTL_MP5)

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::MP5::MCA_CTL_MP5 register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG_CTL. Does not affect error detection, correction, or logging.

_ccd[11:0]_instMP5_n[11:0]_aliasMSRLEGACY; MSR0000_043C

_ccd[11:0]_instMP5_n[11:0]_aliasMSR; MSRC000_20F0

Bits	Description
63:11	Reserved.
10	Mp5FuseSramError. Read-write. Reset: 0. Fuse SRAM ECC or parity error.
9	Mp5ITagBError. Read-write. Reset: 0. Instruction Tag Cache Bank B ECC or parity error.
8	Mp5ITagAError. Read-write. Reset: 0. Instruction Tag Cache Bank A ECC or parity error.
7	Mp5ICacheBError. Read-write. Reset: 0. Instruction Cache Bank B ECC or parity error.
6	Mp5ICacheAError. Read-write. Reset: 0. Instruction Cache Bank A ECC or parity error.
5	Mp5DTagBError. Read-write. Reset: 0. Data Tag Cache Bank B ECC or parity error.
4	Mp5DTagAError. Read-write. Reset: 0. Data Tag Cache Bank A ECC or parity error.
3	Mp5DCacheBError. Read-write. Reset: 0. Data Cache Bank B ECC or parity error.
2	Mp5DCacheAError. Read-write. Reset: 0. Data Cache Bank A ECC or parity error.
1	Mp5LowSramError. Read-write. Reset: 0. Low SRAM ECC or parity error.
0	Mp5HighSramError. Read-write. Reset: 0. High SRAM ECC or parity error.

MSR0000_043D...MSRC000_20F1 [MP5 Machine Check Status] (MCA::MP5::MCA_STATUS_MP5)

Reset: Cold,0000_0000_0000_0000h.

Logs information associated with errors.

_ccd[11:0]_instMP5_n[11:0]_aliasMSRLEGACY; MSR0000_043D

_ccd[11:0]_instMP5_n[11:0]_aliasMSR; MSRC000_20F1

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::MP5::MCA_CTL_MP5. This bit is a copy of bit in MCA::MP5::MCA_CTL_MP5 for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	MiscV. Reset: Cold,0. 1=Valid thresholding in MCA::MP5::MCA_MISC0_MP5. 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	AddrV. Reset: Cold,0. 1=MCA::MP5::MCA_ADDR_MP5 contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	PCC. 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	ErrCoreIdVal. 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	TCC. 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::MP5::MCA_STATUS_MP5[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	RESERV54. 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	SyndV. Reset: Cold,0. 1=This error logged information in MCA::MP5::MCA_SYND_MP5. If MCA::MP5::MCA_SYND_MP5[ErrorPriority] is the same as the priority of the error in MCA::MP5::MCA_STATUS_MP5, then the information in MCA::MP5::MCA_SYND_MP5 is associated with the error in MCA::MP5::MCA_STATUS_MP5. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::MP5::MCA_ADDR_MP5[ErrorAddr]. A value of 0 indicates that MCA::MP5::MCA_ADDR_MP5[63:0] contains a valid byte address. A value of 6 indicates that MCA::MP5::MCA_ADDR_MP5[63:6] contains a valid cache line address and that MCA::MP5::MCA_ADDR_MP5[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::MP5::MCA_ADDR_MP5[63:12] contain a valid 4KB memory page and that MCA::MP5::MCA_ADDR_MP5[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	RESERV22. 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	ErrorCodeExt. 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::MP5::MCA_CTL_MP5 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	ErrorCode. 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 88: MCA_STATUS_MP5

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
Mp5HighSramError	0x0	0/1	0/1	0/1	0	0	1
Mp5LowSramError	0x1	0/1	0/1	0/1	0	0	1
Mp5DCacheAError	0x2	0/1	0/1	0/1	0	0	1
Mp5DCacheBError	0x3	0/1	0/1	0/1	0	0	1
Mp5DTagAError	0x4	0/1	0/1	0/1	0	0	1
Mp5DTagB	0x5	0/1	0/1	0/1	0	0	1

Error							
Mp5ICache AError	0x6	0/1	0/1	0/1	0	0	1
Mp5ICache BError	0x7	0/1	0/1	0/1	0	0	1
Mp5ITagAE rror	0x8	0/1	0/1	0/1	0	0	1
Mp5ITagBEr ror	0x9	0/1	0/1	0/1	0	0	1
Mp5FuseSra mError	0xa	0/1	0/1	0/1	0	0	1

MSR0000_043E...MSRC000_20F2 [MP5 Machine Check Address] (MCA::MP5::MCA_ADDR_MP5)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

MCA::MP5::MCA_ADDR_MP5 stores an address and other information associated with the error in MCA::MP5::MCA_STATUS_MP5. The register is only meaningful if MCA::MP5::MCA_STATUS_MP5[Val]=1 and MCA::MP5::MCA_STATUS_MP5[AddrV]=1.

_ccd[11:0]_instMP5_n[11:0]_aliasMSRLEGACY; MSR0000_043E

_ccd[11:0]_instMP5_n[11:0]_aliasMSR; MSRC000_20F2

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::MP5::MCA_STATUS_MP5. For physical addresses, the most significant bit is given by Core::X86::CpuId::LongModeInfo[PhysAddrSize].

Table 89: MCA_ADDR_MP5

Error Type	Bits	Description
Mp5HighSramError	[55:0]	Reserved
Mp5LowSramError	[55:0]	Reserved
Mp5DCacheAError	[55:0]	Reserved
Mp5DCacheBError	[55:0]	Reserved
Mp5DTagAError	[55:0]	Reserved
Mp5DTagBError	[55:0]	Reserved
Mp5ICacheAError	[55:0]	Reserved
Mp5ICacheBError	[55:0]	Reserved
Mp5ITagAError	[55:0]	Reserved
Mp5ITagBError	[55:0]	Reserved
Mp5FuseSramError	[55:0]	Reserved

MSR0000_043F...MSRC000_20F3 [MP5 Machine Check Miscellaneous 0] (MCA::MP5::MCA_MISC0_MP5)

Log miscellaneous information associated with errors.

_ccd[11:0]_instMP5_n[11:0]_aliasMSRLEGACY; MSR0000_043F

_ccd[11:0]_instMP5_n[11:0]_aliasMSR; MSRC000_20F3

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr:: HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr:: HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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::MP5::MCA_MISC0_MP5[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	LvtOffset. 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::MP5::MCA_MISC0_MP5[Locked]) ? Read-write : Read-only.
51	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr:: HWCR[McStatusWrEn] !MCA::MP5::MCA_MISC0_MP5[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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::MP5::MCA_MISC0_MP5[Locked]) ? Read-write : Read-only.
48	Ovrflw. 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::MP5::MCA_MISC0_MP5[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	ErrCnt. 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::MP5::MCA_MISC0_MP5[Locked]) ? Read-write : Read-only.
31:24	BlkPtr. Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

MSRC000_20F4 [MP5 Machine Check Configuration] (MCA::MP5::MCA_CONFIG_MP5)

Reset: 0000_0000_0000_0121h.

Controls configuration of the associated machine check bank.

_ccd[11:0]_instMP5_n[11:0]_aliasMSR; MSRC000_20F4

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	McaXEnable. 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	IntPresent. Read-only, Volatile . Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::MP5::MCA_CONFIG_MP5[IntEn].
9	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::MP5::MCA_CONFIG_MP5[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. Read-only. Reset: 1. 1=MCA::MP5::MCA_CONFIG_MP5[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::MP5::MCA_CONFIG_MP5[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	DeferredErrorLoggingSupported. 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	McaX. Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::MP5::MCA_MISC0_MP5[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::MP5::MCA_STATUS_MP5[TCC] is present.

MSRC000_20F5 [MP5 IP Identification] (MCA::MP5::MCA_IPID_MP5)

Reset: 0002_0001_0000_0000h.

The MCA::MP5::MCA_IPID_MP5 register is used by software to determine what IP type and revision is associated with the MCA bank.

_ccd[11:0]_instMP5_n[11:0]_aliasMSR; MSRC000_20F5

Bits	Description
63:48	McaType . Read-only. Reset: 0002h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi . 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	HardwareID . Read-only. Reset: 001h. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _ccd0_instMP5_n0_aliasMSR: 3043_0400h
	Init: _ccd1_instMP5_n1_aliasMSR: 3243_0400h
	Init: _ccd2_instMP5_n2_aliasMSR: 3443_0400h
	Init: _ccd3_instMP5_n3_aliasMSR: 3643_0400h
	Init: _ccd4_instMP5_n4_aliasMSR: 3843_0400h
	Init: _ccd5_instMP5_n5_aliasMSR: 3A43_0400h
	Init: _ccd6_instMP5_n6_aliasMSR: 3C43_0400h
	Init: _ccd7_instMP5_n7_aliasMSR: 3E43_0400h
	Init: _ccd8_instMP5_n8_aliasMSR: 4A43_0400h
	Init: _ccd9_instMP5_n9_aliasMSR: 4C43_0400h
	Init: _ccd10_instMP5_n10_aliasMSR: 4E43_0400h
	Init: _ccd11_instMP5_n11_aliasMSR: 5043_0400h

MSRC000_20F6 [MP5 Machine Check Syndrome] (MCA::MP5::MCA_SYND_MP5)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::MP5::MCA_STATUS_MP5 [Thread 0](#)

_ccd[11:0]_instMP5_n[11:0]_aliasMSR; MSRC000_20F6

Bits	Description
63:27	Reserved.
26:24	ErrorPriority . Read-write, Volatile . Reset: Cold, 0h. Encodes the priority of the error logged in MCA::MP5::MCA_SYND_MP5. 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	Length . 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	ErrorInformation . Read-write, Volatile . Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 90 [MCA_SYND_MP5].

Table 90: MCA_SYND_MP5

Error Type	Bits	Description
Mp5HighSramError	[17:15] [14:0]	Reserved Reserved
Mp5LowSramError	[17:15] [14:0]	Reserved Reserved
Mp5DCacheAError	[17:8] [7:0]	Reserved Reserved
Mp5DCacheBError	[17:8] [7:0]	Reserved Reserved
Mp5DTagAError	[17:7] [6:0]	Reserved Reserved
Mp5DTagBError	[17:7] [6:0]	Reserved Reserved

Mp5ICacheAError	[17:8] [7:0]	Reserved Reserved
Mp5ICacheBError	[17:8] [7:0]	Reserved Reserved
Mp5ITagAError	[17:6] [5:0]	Reserved Reserved
Mp5ITagBError	[17:6] [5:0]	Reserved Reserved
Mp5FuseSramError	[17:15] [14:0]	Reserved Reserved

MSRC001_040F [MP5 Machine Check Control Mask] (MCA::MP5::MCA_CTL_MASK_MP5)

Read-write. Reset: 0000_0000_0000_0000h.

Inhibit detection of an error source.

_ccd[11:0]_instMP5_n[11:0]_aliasMSR; MSRC001_040F

Bits	Description
63:11	Reserved.
10	Mp5FuseSramError. Read-write. Reset: 0. Fuse SRAM ECC or parity error.
9	Mp5ITagBError. Read-write. Reset: 0. Instruction Tag Cache Bank B ECC or parity error.
8	Mp5ITagAError. Read-write. Reset: 0. Instruction Tag Cache Bank A ECC or parity error.
7	Mp5ICacheBError. Read-write. Reset: 0. Instruction Cache Bank B ECC or parity error.
6	Mp5ICacheAError. Read-write. Reset: 0. Instruction Cache Bank A ECC or parity error.
5	Mp5DTagBError. Read-write. Reset: 0. Data Tag Cache Bank B ECC or parity error.
4	Mp5DTagAError. Read-write. Reset: 0. Data Tag Cache Bank A ECC or parity error.
3	Mp5DCacheBError. Read-write. Reset: 0. Data Cache Bank B ECC or parity error.
2	Mp5DCacheAError. Read-write. Reset: 0. Data Cache Bank A ECC or parity error.
1	Mp5LowSramError. Read-write. Reset: 0. Low SRAM ECC or parity error.
0	Mp5HighSramError. Read-write. Reset: 0. High SRAM ECC or parity error.

MSRC000_20FE [MP5 Machine Check Syndrome Extended] (MCA::MP5::MCA_SYND1_MP5)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::MP5::MCA_STATUS_MP5 [Thread 0](#)

_ccd[11:0]_instMP5_n[11:0]_aliasMSR; MSRC000_20FE

Bits	Description
63:0	Syndrome. Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::MP5::MCA_SYND1_MP5 register stores information associated with the error in MCA::MP5::MCA_STATUS_MP5 or MCA_DESTAT. The register is meaningful if MCA::MP5::MCA_STATUS_MP5[SyndV]=1. When MCA::MP5::MCA_CONFIG_MP5[McaFruTextInMca]=1, MCA::MP5::MCA_SYND1_MP5 stores ASCII FruText associated with the error.

MSRC000_20FF [MP5 Machine Check Syndrome Extended] (MCA::MP5::MCA_SYND2_MP5)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::MP5::MCA_STATUS_MP5 [Thread 0](#)

_ccd[11:0]_instMP5_n[11:0]_aliasMSR; MSRC000_20FF

Bits	Description
63:0	Syndrome. Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::MP5::MCA_SYND2_MP5 register stores information associated with the error in MCA::MP5::MCA_STATUS_MP5 or MCA_DESTAT. The register is meaningful if MCA::MP5::MCA_STATUS_MP5[SyndV]=1. When MCA::MP5::MCA_CONFIG_MP5[McaFruTextInMca]=1, MCA::MP5::MCA_SYND2_MP5 stores ASCII FruText associated with the error.

3.2.5.14 NBIO

MSR0000_046C...MSRC000_21B0 [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.

_inst[IOHC[3:0]IOHC]_iohub[3:0]_n[7:0]_nbio[1:0]_aliasMSRLEGACY; MSR0000_046C

_inst[IOHC[3:0]IOHC]_iohub[3:0]_n[7:0]_nbio[1:0]_aliasMSR; MSRC000_21B0

Bits	Description
63:6	Reserved.
5	Int_ErrEvent. Read-write. Reset: 0. Internal system fatal error event was detected.
4	IOHC_Internal_Poison. Read-write. Reset: 0. Internal Poison Error. Poison data was sent to an internal client.
3	Egress_Poison. Read-write. Reset: 0. SDP Egress Poison Error. Poison was propagated to an egress port.
2	Ext_ErrEvent. Read-write. Reset: 0. External SDP ErrEvent error. A system fatal error event from an SDP interface was detected.
1	PCIE_Sideband. Read-write. Reset: 0. PCIE error. A PCIE error was logged in a PCIE root port.
0	EccParityError. Read-write. Reset: 0. ECC or Parity error. An SRAM ECC or parity error was detected.

MSR0000_046D...MSRC000_21B1 [NBIO Machine Check Status] (MCA::NBIO::MCA_STATUS_NBIO)

Reset: Cold,0000_0000_0000_0000h.

Logs information associated with errors.

_inst[IOHC[3:0]IOHC]_iohub[3:0]_n[7:0]_nbio[1:0]_aliasMSRLEGACY; MSR0000_046D

_inst[IOHC[3:0]IOHC]_iohub[3:0]_n[7:0]_nbio[1:0]_aliasMSR; MSRC000_21B1

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. 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	MiscV. 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	AddrV. 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	PCC. 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	ErrCoreIdVal. 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	TCC. 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	RESERV54. 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	SyndV. 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	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. 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	RESERV22. 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	ErrorCodeExt. 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	ErrorCode. 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 91: 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_046E...MSRC000_21B2 [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.

_inst[IOHC[3:0]IOHC]_iohub[3:0]_n[7:0]_nbio[1:0]_aliasMSRLEGACY; MSR0000_046E

_inst[IOHC[3:0]IOHC]_iohub[3:0]_n[7:0]_nbio[1:0]_aliasMSR; MSRC000_21B2

Bits	Description
63:0	ErrorAddr. 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 92: 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_046F...MSRC000_21B3 [NBIO Machine Check Miscellaneous 0] (MCA::NBIO::MCA_MISC0_NBIO)

Log miscellaneous information associated with errors.

_inst[IOHC[3:0]IOHC]_iohub[3:0]_n[7:0]_nbio[1:0]_aliasMSRLEGACY; MSR0000_046F

_inst[IOHC[3:0]IOHC]_iohub[3:0]_n[7:0]_nbio[1:0]_aliasMSR; MSRC000_21B3

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr:: HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr:: HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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	LvtOffset. 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	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr:: HWCR[McStatusWrEn] !MCA::NBIO::MCA_MISC0_NBIO[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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	Ovrlw. 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	ErrCnt. 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	BlkPtr. Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

MSRC000_21B4 [NBIO Machine Check Configuration] (MCA::NBIO::MCA_CONFIG_NBIO)

Reset: 0000_0002_0000_0125h.

Controls configuration of the associated machine check bank.

_inst[IOHC[3:0]IOHC]_iohub[3:0]_n[7:0]_nbio[1:0]_aliasMSR; MSRC000_21B4

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	LogDeferredInMcaStat. 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	McaXEnable. 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	IntPresent. 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	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::NBIO::MCA_CONFIG_NBIO[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. 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	DeferredErrorLoggingSupported. 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	McaX. 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_21B5 [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.

_inst[IOHC[3:0]IOHC]_iohub[3:0]_n[7:0]_nbio[1:0]_aliasMSR; MSRC000_21B5

Bits	Description
63:48	McaType. Read-only. Reset: 0000h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi. 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	HardwareID. Read-only. Reset: 018h. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId. Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register. Init: _instIOHC0IOHC_iohub0_n0_nbio0_aliasMSR: 13B1_7000h Init: _instIOHC0IOHC_iohub0_n2_nbio1_aliasMSR: 13D1_7000h Init: _instIOHC1IOHC_iohub1_n4_nbio0_aliasMSR: 1D41_7000h Init: _instIOHC1IOHC_iohub1_n6_nbio1_aliasMSR: 1D61_7000h Init: _instIOHC2IOHC_iohub2_n1_nbio0_aliasMSR: 13C1_7000h Init: _instIOHC2IOHC_iohub2_n3_nbio1_aliasMSR: 13E1_7000h Init: _instIOHC3IOHC_iohub3_n5_nbio0_aliasMSR: 1D51_7000h Init: _instIOHC3IOHC_iohub3_n7_nbio1_aliasMSR: 1D71_7000h

MSRC000_21B6 [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](#)

_inst[IOHC[3:0]IOHC]_iohub[3:0]_n[7:0]_nbio[1:0]_aliasMSR; MSRC000_21B6

Bits	Description
63:33	Reserved.
32	Syndrom. 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	ErrorPriority. 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	Length. Read-write, Volatile . Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::NBIO::MCA_SYND_NBIO[Syndrome]. Length values greater than 32 (decimal) are interpreted as equal to 32 (decimal). 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	ErrorInformation. Read-write, Volatile . Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 93 [MCA_SYND_NBIO].

Table 93: 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_21B8 [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.

_inst[IOHC[3:0]IOHC]_iohub[3:0]_n[7:0]_nbio[1:0]_aliasMSR; MSRC000_21B8

Bits	Description
63	Val. Read-write, Volatile . Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	Overflow. 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 overwrite priorities.)
61:59	RESERV4. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	AddrV. Read-write, Volatile . Reset: Cold,0. 1=MCA::NBIO::MCA_DEADDR_NBIO contains address information associated with the error.
57:54	RESERV3. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	SyndV. 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	RESERV2. Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	Deferred. 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	RESERV1. Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	AddrLsb. 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	RESERV0. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	ErrorCodeExt. 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	ErrorCode. Read-write, Volatile . Reset: Cold,0000h. Error code for this error.

MSRC000_21B9 [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].

_inst[IOHC[3:0]IOHC]_iohub[3:0]_n[7:0]_nbio[1:0]_aliasMSR; MSRC000_21B9

Bits	Description
63:0	ErrorAddr. 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_041B [NBIO Machine Check Control Mask] (MCA::NBIO::MCA_CTL_MASK_NBIO)

Read-write. Reset: 0000_0000_0000_0000h.

Inhibit detection of an error source.

`_inst[IOHC[3:0]IOHC]_iohub[3:0]_n[7:0]_nbio[1:0]_aliasMSR; MSRC001_041B`

Bits	Description
63:6	Reserved.
5	Int_ErrEvent. Read-write. Reset: 0. Internal system fatal error event was detected.
4	IOHC_Internal_Poison. Read-write. Reset: 0. Internal Poison Error. Poison data was sent to an internal client.
3	Egress_Poison. Read-write. Reset: 0. SDP Egress Poison Error. Poison was propagated to an egress port.
2	Ext_ErrEvent. Read-write. Reset: 0. Init: BIOS,1. External SDP ErrEvent error. A system fatal error event from an SDP interface was detected.
1	PCIE_Sideband. Read-write. Reset: 0. Init: BIOS,1. PCIE error. A PCIE error was logged in a PCIE root port.
0	EccParityError. Read-write. Reset: 0. ECC or Parity error. An SRAM ECC or parity error was detected.

MSRC000_21BE [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](#)`_inst[IOHC[3:0]IOHC]_iohub[3:0]_n[7:0]_nbio[1:0]_aliasMSR; MSRC000_21BE`

Bits	Description
63:0	Syndrome. 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_21BF [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](#)`_inst[IOHC[3:0]IOHC]_iohub[3:0]_n[7:0]_nbio[1:0]_aliasMSR; MSRC000_21BF`

Bits	Description
63:0	Syndrome. 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.

3.2.5.15 PCIE**MSR0000_0464...MSRC000_21A0 [PCIE Machine Check Control] (MCA::PCIE::MCA_CTL_PCIE)**

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::PCIE::MCA_CTL_PCIE register must be enabled by the corresponding enable bit in Core::X86::Msr::[MCG_CTL](#). Does not affect error detection, correction, or logging.

`_inst[PCIE[3:0]PCIE]_n[7:0]_nbio[1:0]_aliasMSRLEGACY; MSR0000_0464``_instPCIE5PCIE_n8_nbio0_aliasMSRLEGACY; MSR0000_0468``_inst[PCIE[3:0]PCIE]_n[7:0]_nbio[1:0]_aliasMSR; MSRC000_2190``_instPCIE5PCIE_n8_nbio0_aliasMSR; MSRC000_21A0`

Bits	Description
63:1	Reserved.
0	SDP_PARITY_ERR_LOG. Read-write. Reset: 0. SDP Data Parity Error logging.

MSR0000_0465...MSRC000_21A1 [PCIE Machine Check Status] (MCA::PCIE::MCA_STATUS_PCIE)

Reset: Cold,0000_0000_0000_0000h.

Logs information associated with errors.

_inst[PCIE[3:0]PCIE]_n[7:0]_nbio[1:0]_aliasMSRLEGACY; MSR0000_0465

_instPCIE5PCIE_n8_nbio0_aliasMSRLEGACY; MSR0000_0469

_inst[PCIE[3:0]PCIE]_n[7:0]_nbio[1:0]_aliasMSR; MSRC000_2191

_instPCIE5PCIE_n8_nbio0_aliasMSR; MSRC000_21A1

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::PCIE::MCA_CTL_PCIE. This bit is a copy of bit in MCA::PCIE::MCA_CTL_PCIE for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	MiscV. Reset: Cold,0. 1=Valid thresholding in MCA::PCIE::MCA_MISC0_PCIE. 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	AddrV. Reset: Cold,0. 1=MCA::PCIE::MCA_ADDR_PCIE contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	PCC. 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	ErrCoreIdVal. 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	TCC. 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::PCIE::MCA_STATUS_PCIE[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	RESERV54. 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	SyndV. Reset: Cold,0. 1=This error logged information in MCA::PCIE::MCA_SYND_PCIE. If MCA::PCIE::MCA_SYND_PCIE[ErrorPriority] is the same as the priority of the error in MCA::PCIE::MCA_STATUS_PCIE, then the information in MCA::PCIE::MCA_SYND_PCIE is associated with the error in MCA::PCIE::MCA_STATUS_PCIE. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::PCIE::MCA_ADDR_PCIE[ErrorAddr]. A value of 0 indicates that MCA::PCIE::MCA_ADDR_PCIE[63:0] contains a valid byte address. A value of 6 indicates that MCA::PCIE::MCA_ADDR_PCIE[63:6] contains a valid cache line address and that MCA::PCIE::MCA_ADDR_PCIE[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::PCIE::MCA_ADDR_PCIE[63:12] contain a valid 4KB memory page and that MCA::PCIE::MCA_ADDR_PCIE[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	RESERV22. 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	ErrorCodeExt. 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::PCIE::MCA_CTL_PCIE 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	ErrorCode. 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 94: MCA_STATUS_PCIE

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
SDP_PARITY_ERR_LOG	0x0	0	0	0	1	0	1

MSR0000_0466...MSRC000_21A2 [PCIE Machine Check Address] (MCA::PCIE::MCA_ADDR_PCIE)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

MCA::PCIE::MCA_ADDR_PCIE stores an address and other information associated with the error in MCA::PCIE::MCA_STATUS_PCIE. The register is only meaningful if MCA::PCIE::MCA_STATUS_PCIE[Val]=1 and MCA::PCIE::MCA_STATUS_PCIE[AddrV]=1.

_inst[PCIE[3:0]PCIE]_n[7:0]_nbio[1:0]_aliasMSRLEGACY; MSR0000_0466

_instPCIE5PCIE_n8_nbio0_aliasMSRLEGACY; MSR0000_046A

_inst[PCIE[3:0]PCIE]_n[7:0]_nbio[1:0]_aliasMSR; MSRC000_2192

_instPCIE5PCIE_n8_nbio0_aliasMSR; MSRC000_21A2

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::PCIE::MCA_STATUS_PCIE. For physical addresses, the most significant bit is given by Core::X86::Cpuid::LongModeInfo[PhysAddrSize].

Table 95: MCA_ADDR_PCIE

Error Type	Bits	Description
SDP_PARITY_ERR_LOG	[63:6]	MST_PCIE_ReqAddr[63:6]
	[5:0]	Reserved

MSR0000_0467...MSRC000_21A3 [PCIE Machine Check Miscellaneous 0] (MCA::PCIE::MCA_MISC0_PCIE)

Log miscellaneous information associated with errors.

[_inst\[PCIE\[3:0\]PCIE\]_n\[7:0\]_nbio\[1:0\]_aliasMSRLEGACY; MSR0000_0467](#)[_instPCIE5PCIE_n8_nbio0_aliasMSRLEGACY; MSR0000_046B](#)[_inst\[PCIE\[3:0\]PCIE\]_n\[7:0\]_nbio\[1:0\]_aliasMSR; MSRC000_2193](#)[_instPCIE5PCIE_n8_nbio0_aliasMSR; MSRC000_21A3](#)

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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::PCIE::MCA_MISC0_PCIE[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	LvtOffset. 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::PCIE::MCA_MISC0_PCIE[Locked]) ? Read-write : Read-only.
51	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msrr::HWCR[McStatusWrEn] !MCA::PCIE::MCA_MISC0_PCIE[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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::PCIE::MCA_MISC0_PCIE[Locked]) ? Read-write : Read-only.
48	Ovrflw. 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::PCIE::MCA_MISC0_PCIE[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	ErrCnt. 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::PCIE::MCA_MISC0_PCIE[Locked]) ? Read-write : Read-only.
31:24	BlkPtr. Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

MSRC000_21[9...A]4 [PCIE Machine Check Configuration] (MCA::PCIE::MCA_CONFIG_PCIE)

Reset: 0000_0002_0000_0125h.

Controls configuration of the associated machine check bank.

_inst[PCIE[3:0]PCIE]_n[7:0]_nbio[1:0]_aliasMSR; MSRC000_2194

_instPCIE5PCIE_n8_nbio0_aliasMSR; MSRC000_21A4

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	LogDeferredInMcaStat. Read-write. Reset: 0. Init: BIOS,1. 1=Log deferred errors in MCA::PCIE::MCA_STATUS_PCIE and MCA::PCIE::MCA_ADDR_PCIE in addition to MCA::PCIE::MCA_DESTAT_PCIE and MCA::PCIE::MCA_DEADDR_PCIE. 0=Only log deferred errors in MCA::PCIE::MCA_DESTAT_PCIE and MCA::PCIE::MCA_DEADDR_PCIE. This bit does not affect logging of deferred errors in MCA::PCIE::MCA_SYND_PCIE, MCA::PCIE::MCA_MISC0_PCIE.
33	Reserved.
32	McaXEnable. 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	IntPresent. Read-only, Volatile . Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::PCIE::MCA_CONFIG_PCIE[IntEn].
9	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::PCIE::MCA_CONFIG_PCIE[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. Read-only. Reset: 1. 1=MCA::PCIE::MCA_CONFIG_PCIE[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::PCIE::MCA_CONFIG_PCIE[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	DeferredErrorLoggingSupported. Read-only. Reset: 1. 1=Deferred errors are supported in this MCA bank, and MCA::PCIE::MCA_CONFIG_PCIE[LogDeferredInMcaStat] controls the logging behavior of these errors. MCA::PCIE::MCA_DESTAT_PCIE and MCA::PCIE::MCA_DEADDR_PCIE are supported in this MCA bank. 0=Deferred errors are not supported in this bank.
1	Reserved.
0	McaX. Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::PCIE::MCA_MISC0_PCIE[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::PCIE::MCA_STATUS_PCIE[TCC] is present.

MSRC000_21[9...A]5 [PCIE IP Identification] (MCA::PCIE::MCA_IPID_PCIE)

Reset: 0001_0046_0000_0000h.

The MCA::PCIE::MCA_IPID_PCIE register is used by software to determine what IP type and revision is associated with the MCA bank.

_inst[PCIE[3:0]PCIE]_n[7:0]_nbio[1:0]_aliasMSR; MSRC000_2195

_instPCIE5PCIE_n8_nbio0_aliasMSR; MSRC000_21A5

Bits	Description
63:48	McaType . Read-only. Reset: 0001h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi . 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	HardwareID . Read-only. Reset: 046h. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _instPCIE0PCIE_n0_nbio0_aliasMSR: 1A3C_0000h
	Init: _instPCIE0PCIE_n1_nbio1_aliasMSR: 1A5C_0000h
	Init: _instPCIE1PCIE_n2_nbio0_aliasMSR: 1A4C_0000h
	Init: _instPCIE1PCIE_n3_nbio1_aliasMSR: 1A6C_0000h
	Init: _instPCIE2PCIE_n4_nbio0_aliasMSR: 1A7C_0000h
	Init: _instPCIE2PCIE_n5_nbio1_aliasMSR: 1A9C_0000h
	Init: _instPCIE3PCIE_n6_nbio0_aliasMSR: 1A8C_0000h
	Init: _instPCIE3PCIE_n7_nbio1_aliasMSR: 1AAC_0000h
	Init: _instPCIE5PCIE_n8_nbio0_aliasMSR: 1ABC_0000h

MSRC000_21[9...A]6 [PCIE Machine Check Syndrome] (MCA::PCIE::MCA_SYND_PCIE)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::PCIE::MCA_STATUS_PCIE [Thread 0](#)

_inst[PCIE[3:0]PCIE]_n[7:0]_nbio[1:0]_aliasMSR; MSRC000_2196

_instPCIE5PCIE_n8_nbio0_aliasMSR; MSRC000_21A6

Bits	Description
63:32	Syndrom . Read-write, Volatile . Reset: Cold, 0000_0000h. Contains the syndrome, if any, associated with the error logged in MCA::PCIE::MCA_STATUS_PCIE. The low-order bit of the syndrome is stored in bit 0, and the syndrome has a length specified by MCA::PCIE::MCA_SYND_PCIE[Length]. The Syndrome field is only valid when MCA::PCIE::MCA_SYND_PCIE[Length] is not 0.
31:27	Reserved.
26:24	ErrorPriority . Read-write, Volatile . Reset: Cold, 0h. Encodes the priority of the error logged in MCA::PCIE::MCA_SYND_PCIE. 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	Length . Read-write, Volatile . Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::PCIE::MCA_SYND_PCIE[Syndrome]. Length values greater than 32 (decimal) are interpreted as equal to 32 (decimal). A value of 0 indicates that there is no valid syndrome in MCA::PCIE::MCA_SYND_PCIE. For example, a syndrome length of 9 means that MCA::PCIE::MCA_SYND_PCIE[Syndrome] bits [8:0] contains a valid syndrome.
17:0	ErrorInformation . Read-write, Volatile . Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 96 [MCA_SYND_PCIE].

Table 96: MCA_SYND_PCIE

Error Type	Bits	Description
SDP_PARITY_ERR_LOG	[17:11]	Reserved
	[10:1]	ReqTag
	[0]	MST_PCIE or PCIE_SLV interface

MSRC000_21[9...A]8 [PCIE Machine Check Deferred Error Status] (MCA::PCIE::MCA_DESTAT_PCIE)

Reset: Cold,0000_0000_0000_0000h.

Holds status information for the first deferred error seen in this bank.

_inst[PCIE[3:0]PCIE]_n[7:0]_nbio[1:0]_aliasMSR; MSRC000_2198

_instPCIE5PCIE_n8_nbio0_aliasMSR; MSRC000_21A8

Bits	Description
63	Val. Read-write, Volatile . Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	Overflow. 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 overwrite priorities.)
61:59	RESERV4. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	AddrV. Read-write, Volatile . Reset: Cold,0. 1=MCA::PCIE::MCA_DEADDR_PCIE contains address information associated with the error.
57:54	RESERV3. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	SyndV. Read-write, Volatile . Reset: Cold,0. 1=This error logged information in MCA::PCIE::MCA_SYND_PCIE. If MCA::PCIE::MCA_SYND_PCIE[ErrorPriority] is the same as the priority of the error in MCA::PCIE::MCA_STATUS_PCIE, then the information in MCA::PCIE::MCA_SYND_PCIE is associated with the error in MCA::PCIE::MCA_DESTAT_PCIE.
52:45	RESERV2. Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	Deferred. 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	RESERV1. Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	AddrLsb. Read-write, Volatile . Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::PCIE::MCA_ADDR_PCIE[ErrorAddr]. A value of 0 indicates that MCA::PCIE::MCA_ADDR_PCIE[63:0] contains a valid byte address. A value of 6 indicates that MCA::PCIE::MCA_ADDR_PCIE[63:6] contains a valid cache line address and that MCA::PCIE::MCA_ADDR_PCIE[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::PCIE::MCA_ADDR_PCIE[63:12] contain a valid 4KB memory page and that MCA::PCIE::MCA_ADDR_PCIE[11:0] should be ignored by error handling software.
23:22	RESERV0. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	ErrorCodeExt. 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	ErrorCode. Read-write, Volatile . Reset: Cold,0000h. Error code for this error.

MSRC000_21[9...A]9 [PCIE Deferred Error Address] (MCA::PCIE::MCA_DEADDR_PCIE)Read-write, [Volatile](#). Reset: Cold,0000_0000_0000_0000h.

The MCA::PCIE::MCA_DEADDR_PCIE register stores the address associated with the error in MCA::PCIE::MCA_DESTAT_PCIE. The register is only meaningful if MCA::PCIE::MCA_DESTAT_PCIE[Val]=1 and MCA::PCIE::MCA_DESTAT_PCIE[AddrV]=1. The lowest valid bit of the address is defined by MCA::PCIE::MCA_DESTAT_PCIE[AddrLsb].

_inst[PCIE[3:0]PCIE]_n[7:0]_nbio[1:0]_aliasMSR; MSRC000_2199

_instPCIE5PCIE_n8_nbio0_aliasMSR; MSRC000_21A9

Bits	Description
63:0	ErrorAddr. Read-write, Volatile . Reset: Cold,0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::PCIE::MCA_DESTAT_PCIE. The lowest-order valid bit of the address is specified in MCA::PCIE::MCA_DESTAT_PCIE[AddrLsb].

MSRC001_041[9...A] [PCIE Machine Check Control Mask] (MCA::PCIE::MCA_CTL_MASK_PCIE)

Read-write. Reset: 0000_0000_0000_0000h.

Inhibit detection of an error source.

[_inst\[PCIE\[3:0\]PCIE\]_n\[7:0\]_nbio\[1:0\]_aliasMSR; MSRC001_0419](#)[_instPCIE5PCIE_n8_nbio0_aliasMSR; MSRC001_041A](#)**Bits Description**

63:1 Reserved.

0 **SDP_PARITY_ERR_LOG**. Read-write. Reset: 0. SDP Data Parity Error logging.**MSRC000_21[9...A]E [PCIE Machine Check Syndrome Extended] (MCA::PCIE::MCA_SYND1_PCIE)**

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::PCIE::MCA_STATUS_PCIE [Thread 0](#)[_inst\[PCIE\[3:0\]PCIE\]_n\[7:0\]_nbio\[1:0\]_aliasMSR; MSRC000_219E](#)[_instPCIE5PCIE_n8_nbio0_aliasMSR; MSRC000_21AE](#)**Bits Description**

63:0 **Syndrom**. Read-write, [Volatile](#). Reset: Cold, 0000_0000_0000_0000h. The MCA::PCIE::MCA_SYND1_PCIE register stores information associated with the error in MCA::PCIE::MCA_STATUS_PCIE or MCA_DESTAT. The register is meaningful if MCA::PCIE::MCA_STATUS_PCIE[SyndV]=1. When MCA::PCIE::MCA_CONFIG_PCIE[McaFruTextInMca]=1, MCA::PCIE::MCA_SYND1_PCIE stores ASCII FruText associated with the error.

MSRC000_21[9...A]F [PCIE Machine Check Syndrome Extended] (MCA::PCIE::MCA_SYND2_PCIE)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::PCIE::MCA_STATUS_PCIE [Thread 0](#)[_inst\[PCIE\[3:0\]PCIE\]_n\[7:0\]_nbio\[1:0\]_aliasMSR; MSRC000_219F](#)[_instPCIE5PCIE_n8_nbio0_aliasMSR; MSRC000_21AF](#)**Bits Description**

63:0 **Syndrom**. Read-write, [Volatile](#). Reset: Cold, 0000_0000_0000_0000h. The MCA::PCIE::MCA_SYND2_PCIE register stores information associated with the error in MCA::PCIE::MCA_STATUS_PCIE or MCA_DESTAT. The register is meaningful if MCA::PCIE::MCA_STATUS_PCIE[SyndV]=1. When MCA::PCIE::MCA_CONFIG_PCIE[McaFruTextInMca]=1, MCA::PCIE::MCA_SYND2_PCIE stores ASCII FruText associated with the error.

3.2.5.16 KPX SERDES**MSR0000_047C...MSRC000_21F0 [KPX_SERDES Machine Check Control] (MCA::KPX::SERDES::MCA_CTL_KPX_SERDES)**

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::KPX::SERDES::MCA_CTL_KPX_SERDES register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG_CTL. Does not affect error detection, correction, or logging.

[_inst\[SERDESDP2KPXSERDES, SERDESCP0KPXSERDES, SERDESBP\[3,1\]KPXSERDES, SERDESBG\[2:0\]KPXSERDES, SERDESAG3KPXSERDES\]_n\[7:0\]_aliasMSRLEGACY; MSR0000_047C](#)[_inst\[SERDESDP2KPXSERDES, SERDESCP0KPXSERDES, SERDESBP\[3,1\]KPXSERDES, SERDESBG\[2:0\]KPXSERDES, SERDESAG3KPXSERDES\]_n\[7:0\]_aliasMSR; MSRC000_21F0](#)**Bits Description**

63:4 Reserved.

3 **APB**. Read-write. Reset: 0. PHY APB error2 **ARCData**. Read-write. Reset: 0. ARC data buffer parity error1 **ARCIns**. Read-write. Reset: 0. ARC instruction buffer parity error0 **RAMECC**. Read-write. Reset: 0. RAM ECC Error.

**MSR0000_047D...MSRC000_21F1 [KPX_SERDES Machine Check Status]
(MCA::KPX::SERDES::MCA_STATUS_KPX_SERDES)**

Reset: Cold,0000_0000_0000_0000h.

Logs information associated with errors.

_inst[SERDESDP2KPXSERDES,SERDESCP0KPXSERDES,SERDESBP[3,1]KPXSERDES,SERDESBG[2:0]KPXSERDES,SERDESAG3KPXSERDES]_n[7:0]_aliasMSRLEGACY; MSR0000_047D

_inst[SERDESDP2KPXSERDES,SERDESCP0KPXSERDES,SERDESBP[3,1]KPXSERDES,SERDESBG[2:0]KPXSERDES,SERDESAG3KPXSERDES]_n[7:0]_aliasMSR; MSRC000_21F1

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::KPX::SERDES::MCA_CTL_KPX_SERDES. This bit is a copy of bit in MCA::KPX::SERDES::MCA_CTL_KPX_SERDES for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	MiscV. Reset: Cold,0. 1=Valid thresholding in MCA::KPX::SERDES::MCA_MISC0_KPX_SERDES. 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	AddrV. Reset: Cold,0. 1=MCA::KPX::SERDES::MCA_ADDR_KPX_SERDES contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	PCC. 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	ErrCoreIdVal. 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	TCC. 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::KPX::SERDES::MCA_STATUS_KPX_SERDES[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	RESERV54. 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	SyndV. Reset: Cold,0. 1=This error logged information in MCA::KPX::SERDES::MCA_SYND_KPX_SERDES. If MCA::KPX::SERDES::MCA_SYND_KPX_SERDES[ErrorPriority] is the same as the priority of the error in MCA::KPX::SERDES::MCA_STATUS_KPX_SERDES, then the information in MCA::KPX::SERDES::MCA_SYND_KPX_SERDES is associated with the error in MCA::KPX::SERDES::MCA_STATUS_KPX_SERDES. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	RESERV47. 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	CECC . 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	UECC . 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	Deferred . 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	Poison . 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	RESERV41 . 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	Scrub . 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	RESERV38 . 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	ErrCoreId . 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	RESERV30 . 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	AddrLsb . Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::KPX::SERDES::MCA_ADDR_KPX_SERDES[ErrorAddr]. A value of 0 indicates that MCA::KPX::SERDES::MCA_ADDR_KPX_SERDES[63:0] contains a valid byte address. A value of 6 indicates that MCA::KPX::SERDES::MCA_ADDR_KPX_SERDES[63:6] contains a valid cache line address and that MCA::KPX::SERDES::MCA_ADDR_KPX_SERDES[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::KPX::SERDES::MCA_ADDR_KPX_SERDES[63:12] contain a valid 4KB memory page and that MCA::KPX::SERDES::MCA_ADDR_KPX_SERDES[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	RESERV22 . 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	ErrorCodeExt . 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::KPX::SERDES::MCA_CTL_KPX_SERDES 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	ErrorCode . 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 97: MCA_STATUS_KPX_SERDES

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
RAMECC	0x0	0/1	0/1	0/1	0	0	0
ARCIns	0x1	1	1	1	0	0	0
ARCDData	0x2	1	1	1	0	0	0
APB	0x3	1	1	1	0	0	0

**MSR0000_047E...MSRC000_21F2 [KPX_SERDES Machine Check Address]
(MCA::KPX::SERDES::MCA_ADDR_KPX_SERDES)**

Read-only. Reset: Cold,0000_0000_0000_0000h.

MCA::KPX::SERDES::MCA_ADDR_KPX_SERDES stores an address and other information associated with the error in MCA::KPX::SERDES::MCA_STATUS_KPX_SERDES. The register is only meaningful if

MCA::KPX::SERDES::MCA_STATUS_KPX_SERDES[Val]=1 and

MCA::KPX::SERDES::MCA_STATUS_KPX_SERDES[AddrV]=1.

_inst[SERDESDP2KPXSERDES,SERDESCP0KPXSERDES,SERDESBP[3,1]KPXSERDES,SERDESBG[2:0]KPXSERDES,SERDESAG3KPXSERDES]_n[7:0]_a
liasMSRLEGACY; MSR0000_047E

_inst[SERDESDP2KPXSERDES,SERDESCP0KPXSERDES,SERDESBP[3,1]KPXSERDES,SERDESBG[2:0]KPXSERDES,SERDESAG3KPXSERDES]_n[7:0]_a
liasMSR; MSRC000_21F2

Bits	Description
63:0	ErrorAddr. Read-only. Reset: Cold,0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::KPX::SERDES::MCA_STATUS_KPX_SERDES.

Table 98: MCA_ADDR_KPX_SERDES

Error Type	Bits	Description
RAMECC	[63:0]	Reserved
ARCIns	[63:0]	Reserved
ARCDData	[63:0]	Reserved
APB	[63:0]	Reserved

**MSR0000_047F...MSRC000_21F3 [KPX_SERDES Machine Check Miscellaneous 0]
(MCA::KPX::SERDES::MCA_MISC0_KPX_SERDES)**

Log miscellaneous information associated with errors.

_inst[SERDESDP2KPXSERDES,SERDESCP0KPXSERDES,SERDESBP[3,1]KPXSERDES,SERDESBG[2:0]KPXSERDES,SERDESAG3KPXSERDES]_n[7:0]_aliasMSRLEGACY; MSR0000_047F
_inst[SERDESDP2KPXSERDES,SERDESCP0KPXSERDES,SERDESBP[3,1]KPXSERDES,SERDESBG[2:0]KPXSERDES,SERDESAG3KPXSERDES]_n[7:0]_aliasMSR; MSRC000_21F3

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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::KPX::SERDES::MCA_MISC0_KPX_SERDES[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	LvtOffset. 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::KPX::SERDES::MCA_MISC0_KPX_SERDES[Locked]) ? Read-write : Read-only.
51	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn] ! MCA::KPX::SERDES::MCA_MISC0_KPX_SERDES[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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::KPX::SERDES::MCA_MISC0_KPX_SERDES[Locked]) ? Read-write : Read-only.
48	Ovrflw. 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::KPX::SERDES::MCA_MISC0_KPX_SERDES[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	ErrCnt. 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::KPX::SERDES::MCA_MISC0_KPX_SERDES[Locked]) ? Read-write : Read-only.
31:24	BlkPtr. Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

**MSRC000_21F4 [KPX_SERDES Machine Check Configuration]
(MCA::KPX::SERDES::MCA_CONFIG_KPX_SERDES)**

Reset: 0000_0002_0000_0125h.

Controls configuration of the associated machine check bank.

_inst[SERDESDP2KPXSERDES,SERDESCP0KPXSERDES,SERDESBP[3,1]KPXSERDES,SERDESBG[2:0]KPXSERDES,SERDESAG3KPXSERDES]_n[7:0]_aliasMSR; MSRC000_21F4

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	LogDeferredInMcaStat. Read-write. Reset: 0. Init: BIOS,1. 1=Log deferred errors in MCA::KPX::SERDES::MCA_STATUS_KPX_SERDES and MCA::KPX::SERDES::MCA_ADDR_KPX_SERDES in addition to MCA::KPX::SERDES::MCA_DESTAT_KPX_SERDES and MCA::KPX::SERDES::MCA_DEADDR_KPX_SERDES. 0=Only log deferred errors in MCA::KPX::SERDES::MCA_DESTAT_KPX_SERDES and MCA::KPX::SERDES::MCA_DEADDR_KPX_SERDES. This bit does not affect logging of deferred errors in MCA::KPX::SERDES::MCA_SYND_KPX_SERDES, MCA::KPX::SERDES::MCA_MISC0_KPX_SERDES.
33	Reserved.
32	McaXEnable. 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	IntPresent. Read-only, Volatile . Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::KPX::SERDES::MCA_CONFIG_KPX_SERDES[IntEn].
9	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::KPX::SERDES::MCA_CONFIG_KPX_SERDES[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. Read-only. Reset: 1. 1=MCA::KPX::SERDES::MCA_CONFIG_KPX_SERDES[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::KPX::SERDES::MCA_CONFIG_KPX_SERDES[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	DeferredErrorLoggingSupported. Read-only. Reset: 1. 1=Deferred errors are supported in this MCA bank, and MCA::KPX::SERDES::MCA_CONFIG_KPX_SERDES[LogDeferredInMcaStat] controls the logging behavior of these errors. MCA::KPX::SERDES::MCA_DESTAT_KPX_SERDES and MCA::KPX::SERDES::MCA_DEADDR_KPX_SERDES are supported in this MCA bank. 0=Deferred errors are not supported in this bank.
1	Reserved.
0	McaX. Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::KPX::SERDES::MCA_MISC0_KPX_SERDES[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::KPX::SERDES::MCA_STATUS_KPX_SERDES[TCC] is present.

MSRC000_21F5 [KPX_SERDES IP Identification] (MCA::KPX::SERDES::MCA_IPID_KPX_SERDES)

Reset: 0000_0259_0000_0000h.

The MCA::KPX::SERDES::MCA_IPID_KPX_SERDES register is used by software to determine what IP type and revision is associated with the MCA bank.

_inst[SERDESDP2KPXSERDES,SERDESCP0KPXSERDES,SERDESBP[3,1]KPXSERDES,SERDESBG[2:0]KPXSERDES,SERDESAG3KPXSERDES]_n[7:0]_aliasMSR; MSRC000_21F5

Bits	Description
63:48	McaType . Read-only. Reset: 0000h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi . 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	HardwareID . Read-only. Reset: 259h. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _instSERDESAG3KPXSERDES_n1_aliasMSR: 1860_0400h
	Init: _instSERDESBG0KPXSERDES_n5_aliasMSR: 1830_0400h
	Init: _instSERDESBG1KPXSERDES_n6_aliasMSR: 1840_0400h
	Init: _instSERDESBG2KPXSERDES_n7_aliasMSR: 1850_0400h
	Init: _instSERDESBP1KPXSERDES_n2_aliasMSR: 1800_0400h
	Init: _instSERDESBP3KPXSERDES_n4_aliasMSR: 1820_0400h
	Init: _instSERDESCP0KPXSERDES_n0_aliasMSR: 17F0_0400h
	Init: _instSERDESDP2KPXSERDES_n3_aliasMSR: 1810_0400h

MSRC000_21F6 [KPX_SERDES Machine Check Syndrome] (MCA::KPX::SERDES::MCA_SYND_KPX_SERDES)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::KPX::SERDES::MCA_STATUS_KPX_SERDES [Thread 0](#)

_inst[SERDESDP2KPXSERDES,SERDESCP0KPXSERDES,SERDESBP[3,1]KPXSERDES,SERDESBG[2:0]KPXSERDES,SERDESAG3KPXSERDES]_n[7:0]_aliasMSR; MSRC000_21F6

Bits	Description
63:33	Reserved.
32	Syndrom . Read-write, Volatile . Reset: Cold, 0. Contains the syndrome, if any, associated with the error logged in MCA::KPX::SERDES::MCA_STATUS_KPX_SERDES. The low-order bit of the syndrome is stored in bit 0, and the syndrome has a length specified by MCA::KPX::SERDES::MCA_SYND_KPX_SERDES[Length]. The Syndrome field is only valid when MCA::KPX::SERDES::MCA_SYND_KPX_SERDES[Length] is not 0.
31:27	Reserved.
26:24	ErrorPriority . Read-write, Volatile . Reset: Cold, 0h. Encodes the priority of the error logged in MCA::KPX::SERDES::MCA_SYND_KPX_SERDES. 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	Length . Read-write, Volatile . Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::KPX::SERDES::MCA_SYND_KPX_SERDES[Syndrome]. Length values greater than 32 (decimal) are interpreted as equal to 32 (decimal). A value of 0 indicates that there is no valid syndrome in MCA::KPX::SERDES::MCA_SYND_KPX_SERDES. For example, a syndrome length of 9 means that MCA::KPX::SERDES::MCA_SYND_KPX_SERDES[Syndrome] bits [8:0] contains a valid syndrome.
17:0	ErrorInformation . Read-write, Volatile . Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 99 [MCA_SYND_KPX_SERDES].

Table 99: MCA_SYND_KPX_SERDES

Error Type	Bits	Description
RAMECC	[17:0]	Reserved
ARCIns	[17:0]	Reserved
ARCDData	[17:0]	Reserved
APB	[17:0]	Reserved

**MSRC000_21F8 [KPX_SERDES Machine Check Deferred Error Status]
(MCA::KPX::SERDES::MCA_DESTAT_KPX_SERDES)**

Reset: Cold,0000_0000_0000_0000h.

Holds status information for the first deferred error seen in this bank.

_inst[SERDESDP2KPXSERDES,SERDESCP0KPXSERDES,SERDESBP[3,1]KPXSERDES,SERDESBG[2:0]KPXSERDES,SERDESAG3KPXSERDES]_n[7:0]_aliasMSR; MSRC000_21F8

Bits	Description
63	Val. Read-write, Volatile . Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	Overflow. 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 overwrite priorities.)
61:59	RESERV4. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	AddrV. Read-write, Volatile . Reset: Cold,0. 1=MCA::KPX::SERDES::MCA_DEADDR_KPX_SERDES contains address information associated with the error.
57:54	RESERV3. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	SyndV. Read-write, Volatile . Reset: Cold,0. 1=This error logged information in MCA::KPX::SERDES::MCA_SYND_KPX_SERDES. If MCA::KPX::SERDES::MCA_SYND_KPX_SERDES[ErrorPriority] is the same as the priority of the error in MCA::KPX::SERDES::MCA_STATUS_KPX_SERDES, then the information in MCA::KPX::SERDES::MCA_SYND_KPX_SERDES is associated with the error in MCA::KPX::SERDES::MCA_DESTAT_KPX_SERDES.
52:45	RESERV2. Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	Deferred. 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	RESERV1. Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	AddrLsb. Read-write, Volatile . Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::KPX::SERDES::MCA_ADDR_KPX_SERDES[ErrorAddr]. A value of 0 indicates that MCA::KPX::SERDES::MCA_ADDR_KPX_SERDES[63:0] contains a valid byte address. A value of 6 indicates that MCA::KPX::SERDES::MCA_ADDR_KPX_SERDES[63:6] contains a valid cache line address and that MCA::KPX::SERDES::MCA_ADDR_KPX_SERDES[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::KPX::SERDES::MCA_ADDR_KPX_SERDES[63:12] contain a valid 4KB memory page and that MCA::KPX::SERDES::MCA_ADDR_KPX_SERDES[11:0] should be ignored by error handling software.
23:22	RESERV0. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	ErrorCodeExt. 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	ErrorCode. Read-write, Volatile . Reset: Cold,0000h. Error code for this error.

**MSRC000_21F9 [KPX_SERDES Deferred Error Address]
(MCA::KPX::SERDES::MCA_DEADDR_KPX_SERDES)**

Read-only. Reset: Cold,0000_0000_0000_0000h.

The MCA::KPX::SERDES::MCA_DEADDR_KPX_SERDES register stores the address associated with the error in MCA::KPX::SERDES::MCA_DESTAT_KPX_SERDES. The register is only meaningful if MCA::KPX::SERDES::MCA_DESTAT_KPX_SERDES[Val]=1 and MCA::KPX::SERDES::MCA_DESTAT_KPX_SERDES[AddrV]=1. The lowest valid bit of the address is defined by MCA::KPX::SERDES::MCA_DESTAT_KPX_SERDES[AddrLsb].

_inst[SERDESDP2KPXSERDES,SERDESCP0KPXSERDES,SERDESBP[3,1]KPXSERDES,SERDESBG[2:0]KPXSERDES,SERDESAG3KPXSERDES]_n[7:0]_aliasMSR; MSRC000_21F9

Bits	Description
63:0	ErrorAddr. Read-only. Reset: Cold,0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::KPX::SERDES::MCA_DESTAT_KPX_SERDES.

**MSRC001_041F [KPX_SERDES Machine Check Control Mask]
(MCA::KPX::SERDES::MCA_CTL_MASK_KPX_SERDES)**

Read-write. Reset: 0000_0000_0000_0000h.

Inhibit detection of an error source.

_inst[SERDESDP2KPXSERDES,SERDESCP0KPXSERDES,SERDESBP[3,1]KPXSERDES,SERDESBG[2:0]KPXSERDES,SERDESAG3KPXSERDES]_n[7:0]_aliasMSR; MSRC001_041F

Bits	Description
63:4	Reserved.
3	APB. Read-write. Reset: 0. PHY APB error
2	ARCData. Read-write. Reset: 0. ARC data buffer parity error
1	ARCIns. Read-write. Reset: 0. ARC instruction buffer parity error
0	RAMECC. Read-write. Reset: 0. RAM ECC Error.

**MSRC000_21FE [KPX_SERDES Machine Check Syndrome Extended]
(MCA::KPX::SERDES::MCA_SYND1_KPX_SERDES)**

Read-write, Volatile. Reset: Cold,0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::KPX::SERDES::MCA_STATUS_KPX_SERDES [Thread 0](#)

_inst[SERDESDP2KPXSERDES,SERDESCP0KPXSERDES,SERDESBP[3,1]KPXSERDES,SERDESBG[2:0]KPXSERDES,SERDESAG3KPXSERDES]_n[7:0]_aliasMSR; MSRC000_21FE

Bits	Description
63:0	Syndrome. Read-write, Volatile . Reset: Cold,0000_0000_0000_0000h. The MCA::KPX::SERDES::MCA_SYND1_KPX_SERDES register stores information associated with the error in MCA::KPX::SERDES::MCA_STATUS_KPX_SERDES or MCA_DESTAT. The register is meaningful if MCA::KPX::SERDES::MCA_STATUS_KPX_SERDES[SyndV]=1. When MCA::KPX::SERDES::MCA_CONFIG_KPX_SERDES[McaFruTextInMca]=1, MCA::KPX::SERDES::MCA_SYND1_KPX_SERDES stores ASCII FruText associated with the error.

MSRC000_21FF [KPX_SERDES Machine Check Syndrome Extended] (MCA::KPX::SERDES::MCA_SYND2_KPX_SERDES)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::KPX::SERDES::MCA_STATUS_KPX_SERDES
[Thread 0](#)

_inst[SERDESDP2KPXSERDES, SERDESCP0KPXSERDES, SERDESBP[3,1]KPXSERDES, SERDESBG[2:0]KPXSERDES, SERDESAG3KPXSERDES]_n[7:0]_aliasMSR; MSRC000_21FF

Bits	Description
63:0	Syndrone. Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::KPX::SERDES::MCA_SYND2_KPX_SERDES register stores information associated with the error in MCA::KPX::SERDES::MCA_STATUS_KPX_SERDES or MCA_DESTAT. The register is meaningful if MCA::KPX::SERDES::MCA_STATUS_KPX_SERDES[SyndV]=1. When MCA::KPX::SERDES::MCA_CONFIG_KPX_SERDES[McaFruTextInMca]=1, MCA::KPX::SERDES::MCA_SYND2_KPX_SERDES stores ASCII FruText associated with the error.

3.2.5.17 KPX GMI

MSR0000_0448...MSRC000_2130 [KPX_GMI Machine Check Control] (MCA::KPX::GMI::MCA_CTL_KPX_GMI)

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::KPX::GMI::MCA_CTL_KPX_GMI register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG_CTL. Does not affect error detection, correction, or logging.

_inst[GMICONTAINER[9:1,11,10:10,0]KPXGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSRLEGACY; MSR0000_0448

_ccd[11:0]_inst[GMICONTAINER[13:12]KPXGMI]_n[35:15,[13,11,9,7,5,3,1]]_aliasMSRLEGACY; MSR0000_044C

_inst[GMICONTAINER[9:1,11,10:10,0]KPXGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2120

_ccd[11:0]_inst[GMICONTAINER[13:12]KPXGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_2130

_inst[GMICONTAINER[17:14]KPXGMI]_aliasSMN; GMICNTR[17:14]KPXx00000400; GMICNTR[17:14]KPX=1E[F:C]0_0000h

Bits	Description
63:4	Reserved.
3	APB. Read-write. Reset: 0. PHY APB error
2	ARCData. Read-write. Reset: 0. ARC data buffer parity error
1	ARCIns. Read-write. Reset: 0. ARC instruction buffer parity error
0	RAMECC. Read-write. Reset: 0. RAM ECC Error.

**MSR0000_0449...MSRC000_2131 [KPX_GMI Machine Check Status]
(MCA::KPX::GMI::MCA_STATUS_KPX_GMI)**

Reset: Cold,0000_0000_0000_0000h.

Logs information associated with errors.

_inst[GMICONTAINER[9:1,11,10:10,0]KPXGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSRLEGACY; MSR0000_0449

_ccd[11:0]_inst[GMICONTAINER[13:12]KPXGMI]_n[35:15,[13,11,9,7,5,3,1]]_aliasMSRLEGACY; MSR0000_044D

_inst[GMICONTAINER[9:1,11,10:10,0]KPXGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2121

_ccd[11:0]_inst[GMICONTAINER[13:12]KPXGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_2131

_inst[GMICONTAINER[17:14]KPXGMI]_aliasSMN; GMICNTR[17:14]KPXx00000408; GMICNTR[17:14]KPX=1E[F:C]0_0000h

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::KPX::GMI::MCA_CTL_KPX_GMI. This bit is a copy of bit in MCA::KPX::GMI::MCA_CTL_KPX_GMI for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	MiscV. Reset: Cold,0. 1=Valid thresholding in MCA::KPX::GMI::MCA_MISC0_KPX_GMI. 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	AddrV. Reset: Cold,0. 1=MCA::KPX::GMI::MCA_ADDR_KPX_GMI contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	PCC. 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	ErrCoreIdVal. 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	TCC. 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::KPX::GMI::MCA_STATUS_KPX_GMI[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	RESERV54. 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	SyndV. Reset: Cold,0. 1=This error logged information in MCA::KPX::GMI::MCA_SYND_KPX_GMI. If MCA::KPX::GMI::MCA_SYND_KPX_GMI[ErrorPriority] is the same as the priority of the error in MCA::KPX::GMI::MCA_STATUS_KPX_GMI, then the information in MCA::KPX::GMI::MCA_SYND_KPX_GMI is associated with the error in MCA::KPX::GMI::MCA_STATUS_KPX_GMI. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	RESERV47. 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	CECC . 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	UECC . 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	Deferred . 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	Poison . 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	RESERV41 . 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	Scrub . 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	RESERV38 . 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	ErrCoreId . 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	RESERV30 . 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	AddrLsb . Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::KPX::GMI::MCA_ADDR_KPX_GMI[ErrorAddr]. A value of 0 indicates that MCA::KPX::GMI::MCA_ADDR_KPX_GMI[63:0] contains a valid byte address. A value of 6 indicates that MCA::KPX::GMI::MCA_ADDR_KPX_GMI[63:6] contains a valid cache line address and that MCA::KPX::GMI::MCA_ADDR_KPX_GMI[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::KPX::GMI::MCA_ADDR_KPX_GMI[63:12] contain a valid 4KB memory page and that MCA::KPX::GMI::MCA_ADDR_KPX_GMI[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	RESERV22 . 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	ErrorCodeExt . 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::KPX::GMI::MCA_CTL_KPX_GMI 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	ErrorCode . 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 100: MCA_STATUS_KPX_GMI

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
RAMECC	0x0	0/1	0/1	0/1	0	0	0
ARCIns	0x1	1	1	1	0	0	0
ARCDData	0x2	1	1	1	0	0	0
APB	0x3	1	1	1	0	0	0

**MSR0000_044A...MSRC000_2132 [KPX_GMI Machine Check Address]
(MCA::KPX::GMI::MCA_ADDR_KPX_GMI)**

Read-only. Reset: Cold,0000_0000_0000_0000h.

MCA::KPX::GMI::MCA_ADDR_KPX_GMI stores an address and other information associated with the error in MCA::KPX::GMI::MCA_STATUS_KPX_GMI. The register is only meaningful if MCA::KPX::GMI::MCA_STATUS_KPX_GMI[Val]=1 and MCA::KPX::GMI::MCA_STATUS_KPX_GMI[AddrV]=1.

_inst[GMICONTAINER[9:1,11,10:10,0]KPXGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSRLEGACY; MSR0000_044A

_ccd[11:0]_inst[GMICONTAINER[13:12]KPXGMI]_n[35:15,[13,11,9,7,5,3,1]]_aliasMSRLEGACY; MSR0000_044E

_inst[GMICONTAINER[9:1,11,10:10,0]KPXGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2122

_ccd[11:0]_inst[GMICONTAINER[13:12]KPXGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_2132

_inst[GMICONTAINER[17:14]KPXGMI]_aliasSMN; GMICNTR[17:14]KPXx00000410; GMICNTR[17:14]KPX=1E[F:C]0_0000h

Bits	Description
63:0	ErrorAddr. Read-only. Reset: Cold,0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::KPX::GMI::MCA_STATUS_KPX_GMI.

Table 101: MCA_ADDR_KPX_GMI

Error Type	Bits	Description
RAMECC	[63:0]	Reserved
ARCIns	[63:0]	Reserved
ARCDData	[63:0]	Reserved
APB	[63:0]	Reserved

**MSR0000_044B...MSRC000_2133 [KPX_GMI Machine Check Miscellaneous 0]
(MCA::KPX::GMI::MCA_MISC0_KPX_GMI)**

Log miscellaneous information associated with errors.

_inst[GMICONTAINER[9:1,11,10:10,0]KPXGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSRLEGACY; MSR0000_044B

_ccd[11:0]_inst[GMICONTAINER[13:12]KPXGMI]_n[35:15,[13,11,9,7,5,3,1]]_aliasMSRLEGACY; MSR0000_044F

_inst[GMICONTAINER[9:1,11,10:10,0]KPXGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2123

_ccd[11:0]_inst[GMICONTAINER[13:12]KPXGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_2133

_inst[GMICONTAINER[17:14]KPXGMI]_aliasSMN; GMICNTR[17:14]KPXx00000418; GMICNTR[17:14]KPX=1E[F:C]0_0000h

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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::KPX::GMI::MCA_MISC0_KPX_GMI[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	LvtOffset. 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::KPX::GMI::MCA_MISC0_KPX_GMI[Locked]) ? Read-write : Read-only.
51	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn] ! MCA::KPX::GMI::MCA_MISC0_KPX_GMI[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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::KPX::GMI::MCA_MISC0_KPX_GMI[Locked]) ? Read-write : Read-only.
48	Ovrflw. 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::KPX::GMI::MCA_MISC0_KPX_GMI[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	ErrCnt. 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::KPX::GMI::MCA_MISC0_KPX_GMI[Locked]) ? Read-write : Read-only.
31:24	BlkPtr. Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

**MSRC000_21[2...3]4 [KPX_GMI Machine Check Configuration]
(MCA::KPX::GMI::MCA_CONFIG_KPX_GMI)**

Reset: 0000_0002_0000_0125h.

Controls configuration of the associated machine check bank.

_inst[GMICONTAINER[9:1,11,10:10,0]KPXGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2124

_ccd[11:0]_inst[GMICONTAINER[13:12]KPXGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_2134

_inst[GMICONTAINER[17:14]KPXGMI]_aliasSMN; GMICNTR[17:14]KPXx00000420; GMICNTR[17:14]KPX=1E[F:C]0_0000h

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	LogDeferredInMcaStat. Read-write. Reset: 0. Init: BIOS,1. 1=Log deferred errors in MCA::KPX::GMI::MCA_STATUS_KPX_GMI and MCA::KPX::GMI::MCA_ADDR_KPX_GMI in addition to MCA::KPX::GMI::MCA_DESTAT_KPX_GMI and MCA::KPX::GMI::MCA_DEADDR_KPX_GMI. 0=Only log deferred errors in MCA::KPX::GMI::MCA_DESTAT_KPX_GMI and MCA::KPX::GMI::MCA_DEADDR_KPX_GMI. This bit does not affect logging of deferred errors in MCA::KPX::GMI::MCA_SYND_KPX_GMI, MCA::KPX::GMI::MCA_MISC0_KPX_GMI.
33	Reserved.
32	McaXEnable. 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	IntPresent. Read-only, Volatile . Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::KPX::GMI::MCA_CONFIG_KPX_GMI[IntEn].
9	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::KPX::GMI::MCA_CONFIG_KPX_GMI[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. Read-only. Reset: 1. 1=MCA::KPX::GMI::MCA_CONFIG_KPX_GMI[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::KPX::GMI::MCA_CONFIG_KPX_GMI[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	DeferredErrorLoggingSupported. Read-only. Reset: 1. 1=Deferred errors are supported in this MCA bank, and MCA::KPX::GMI::MCA_CONFIG_KPX_GMI[LogDeferredInMcaStat] controls the logging behavior of these errors. MCA::KPX::GMI::MCA_DESTAT_KPX_GMI and MCA::KPX::GMI::MCA_DEADDR_KPX_GMI are supported in this MCA bank. 0=Deferred errors are not supported in this bank.
1	Reserved.
0	McaX. Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::KPX::GMI::MCA_MISC0_KPX_GMI[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::KPX::GMI::MCA_STATUS_KPX_GMI[TCC] is present.

MSRC000_21[2...3]5 [KPX_GMI IP Identification] (MCA::KPX::GMI::MCA_IPID_KPX_GMI)

Reset: 0000_0269_0000_0000h.

The MCA::KPX::GMI::MCA_IPID_KPX_GMI register is used by software to determine what IP type and revision is associated with the MCA bank.

_inst[GMICONTAINER[9:1,11,10:10,0]KPXGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2125

_ccd[11:0]_inst[GMICONTAINER[13:12]KPXGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_2135

_inst[GMICONTAINER[17:14]KPXGMI]_aliasSMN; GMICNTR[17:14]KPXx00000428; GMICNTR[17:14]KPX=1E[F:C]0_0000h

Bits	Description
63:48	McaType . Read-only. Reset: 0000h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi . 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	HardwareID . Read-only. Reset: 269h. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _ccd0_instGMICONTAINER12KPXGMI_n1_aliasMSR: 3090_0400h
	Init: _ccd0_instGMICONTAINER13KPXGMI_n24_aliasMSR: 30A0_0400h
	Init: _ccd1_instGMICONTAINER12KPXGMI_n3_aliasMSR: 3290_0400h
	Init: _ccd1_instGMICONTAINER13KPXGMI_n25_aliasMSR: 32A0_0400h
	Init: _ccd2_instGMICONTAINER12KPXGMI_n5_aliasMSR: 3490_0400h
	Init: _ccd2_instGMICONTAINER13KPXGMI_n26_aliasMSR: 34A0_0400h
	Init: _ccd3_instGMICONTAINER12KPXGMI_n7_aliasMSR: 3690_0400h
	Init: _ccd3_instGMICONTAINER13KPXGMI_n27_aliasMSR: 36A0_0400h
	Init: _ccd4_instGMICONTAINER12KPXGMI_n9_aliasMSR: 3890_0400h
	Init: _ccd4_instGMICONTAINER13KPXGMI_n28_aliasMSR: 38A0_0400h
	Init: _ccd5_instGMICONTAINER12KPXGMI_n11_aliasMSR: 3A90_0400h
	Init: _ccd5_instGMICONTAINER13KPXGMI_n29_aliasMSR: 3AA0_0400h
	Init: _ccd6_instGMICONTAINER12KPXGMI_n13_aliasMSR: 3C90_0400h
	Init: _ccd6_instGMICONTAINER13KPXGMI_n30_aliasMSR: 3CA0_0400h
	Init: _ccd7_instGMICONTAINER12KPXGMI_n15_aliasMSR: 3E90_0400h
	Init: _ccd7_instGMICONTAINER13KPXGMI_n31_aliasMSR: 3EA0_0400h
	Init: _ccd8_instGMICONTAINER12KPXGMI_n17_aliasMSR: 4A90_0400h
	Init: _ccd8_instGMICONTAINER13KPXGMI_n32_aliasMSR: 4AA0_0400h
	Init: _ccd9_instGMICONTAINER12KPXGMI_n19_aliasMSR: 4C90_0400h
	Init: _ccd9_instGMICONTAINER13KPXGMI_n33_aliasMSR: 4CA0_0400h
	Init: _ccd10_instGMICONTAINER12KPXGMI_n21_aliasMSR: 4E90_0400h
	Init: _ccd10_instGMICONTAINER13KPXGMI_n34_aliasMSR: 4EA0_0400h
	Init: _ccd11_instGMICONTAINER12KPXGMI_n23_aliasMSR: 5090_0400h
	Init: _ccd11_instGMICONTAINER13KPXGMI_n35_aliasMSR: 50A0_0400h
	Init: _instGMICONTAINER0KPXGMI_n0_aliasMSR: 1890_0400h
	Init: _instGMICONTAINER10KPXGMI_n20_aliasMSR: 19C0_0400h
	Init: _instGMICONTAINER11KPXGMI_n22_aliasMSR: 19D0_0400h
	Init: _instGMICONTAINER1KPXGMI_n2_aliasMSR: 18A0_0400h
	Init: _instGMICONTAINER2KPXGMI_n4_aliasMSR: 18B0_0400h
	Init: _instGMICONTAINER3KPXGMI_n6_aliasMSR: 18C0_0400h
	Init: _instGMICONTAINER4KPXGMI_n8_aliasMSR: 18D0_0400h
	Init: _instGMICONTAINER5KPXGMI_n10_aliasMSR: 1950_0400h
	Init: _instGMICONTAINER6KPXGMI_n12_aliasMSR: 1960_0400h
	Init: _instGMICONTAINER7KPXGMI_n14_aliasMSR: 1970_0400h
	Init: _instGMICONTAINER8KPXGMI_n16_aliasMSR: 1980_0400h
	Init: _instGMICONTAINER9KPXGMI_n18_aliasMSR: 1990_0400h

MSRC000_21[2...3]6 [KPX_GMI Machine Check Syndrome] (MCA::KPX::GMI::MCA_SYND_KPX_GMI)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::KPX::GMI::MCA_STATUS_KPX_GMI [Thread 0](#)

_inst[GMICONTAINER[9:1,11,10:10,0]KPXGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2126

_ccd[11:0]_inst[GMICONTAINER[13:12]KPXGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_2136

_inst[GMICONTAINER[17:14]KPXGMI]_aliasSMN; GMICNTR[17:14]KPXx00000430; GMICNTR[17:14]KPX=1E[F:C]0_0000h

Bits	Description
63:33	Reserved.
32	Syndrone. Read-write, Volatile . Reset: Cold, 0. Contains the syndrome, if any, associated with the error logged in MCA::KPX::GMI::MCA_STATUS_KPX_GMI. The low-order bit of the syndrome is stored in bit 0, and the syndrome has a length specified by MCA::KPX::GMI::MCA_SYND_KPX_GMI[Length]. The Syndrome field is only valid when MCA::KPX::GMI::MCA_SYND_KPX_GMI[Length] is not 0.
31:27	Reserved.
26:24	ErrorPriority. Read-write, Volatile . Reset: Cold, 0h. Encodes the priority of the error logged in MCA::KPX::GMI::MCA_SYND_KPX_GMI. 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	Length. Read-write, Volatile . Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::KPX::GMI::MCA_SYND_KPX_GMI[Syndrome]. Length values greater than 32 (decimal) are interpreted as equal to 32 (decimal). A value of 0 indicates that there is no valid syndrome in MCA::KPX::GMI::MCA_SYND_KPX_GMI. For example, a syndrome length of 9 means that MCA::KPX::GMI::MCA_SYND_KPX_GMI[Syndrome] bits [8:0] contains a valid syndrome.
17:0	ErrorInformation. Read-write, Volatile . Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 102 [MCA_SYND_KPX_GMI].

Table 102: MCA_SYND_KPX_GMI

Error Type	Bits	Description
RAMECC	[17:0]	Reserved
ARCIns	[17:0]	Reserved
ARCDData	[17:0]	Reserved
APB	[17:0]	Reserved

MSRC000_21[2...3]8 [KPX_GMI Machine Check Deferred Error Status] (MCA::KPX::GMI::MCA_DESTAT_KPX_GMI)

Reset: Cold,0000_0000_0000_0000h.

Holds status information for the first deferred error seen in this bank.

_inst[GMICONTAINER[9:1,11,10:10,0]KPXGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2128

_ccd[11:0]_inst[GMICONTAINER[13:12]KPXGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_2138

_inst[GMICONTAINER[17:14]KPXGMI]_aliasSMN; GMICNTR[17:14]KPXx00000440; GMICNTR[17:14]KPX=1E[F:C]0_0000h

Bits	Description
63	Val. Read-write, Volatile . Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	Overflow. 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 overwrite priorities.)
61:59	RESERV4. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	AddrV. Read-write, Volatile . Reset: Cold,0. 1=MCA::KPX::GMI::MCA_DEADDR_KPX_GMI contains address information associated with the error.
57:54	RESERV3. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	SyndV. Read-write, Volatile . Reset: Cold,0. 1=This error logged information in MCA::KPX::GMI::MCA_SYND_KPX_GMI. If MCA::KPX::GMI::MCA_SYND_KPX_GMI[ErrorPriority] is the same as the priority of the error in MCA::KPX::GMI::MCA_STATUS_KPX_GMI, then the information in MCA::KPX::GMI::MCA_SYND_KPX_GMI is associated with the error in MCA::KPX::GMI::MCA_DESTAT_KPX_GMI.
52:45	RESERV2. Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	Deferred. 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	RESERV1. Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	AddrLsb. Read-write, Volatile . Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::KPX::GMI::MCA_ADDR_KPX_GMI[ErrorAddr]. A value of 0 indicates that MCA::KPX::GMI::MCA_ADDR_KPX_GMI[63:0] contains a valid byte address. A value of 6 indicates that MCA::KPX::GMI::MCA_ADDR_KPX_GMI[63:6] contains a valid cache line address and that MCA::KPX::GMI::MCA_ADDR_KPX_GMI[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::KPX::GMI::MCA_ADDR_KPX_GMI[63:12] contain a valid 4KB memory page and that MCA::KPX::GMI::MCA_ADDR_KPX_GMI[11:0] should be ignored by error handling software.
23:22	RESERV0. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	ErrorCodeExt. 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	ErrorCode. Read-write, Volatile . Reset: Cold,0000h. Error code for this error.

MSRC000_21[2...3]9 [KPX_GMI Deferred Error Address] (MCA::KPX::GMI::MCA_DEADDR_KPX_GMI)

Read-only. Reset: Cold,0000_0000_0000_0000h.

The MCA::KPX::GMI::MCA_DEADDR_KPX_GMI register stores the address associated with the error in MCA::KPX::GMI::MCA_DESTAT_KPX_GMI. The register is only meaningful if MCA::KPX::GMI::MCA_DESTAT_KPX_GMI[Val]=1 and MCA::KPX::GMI::MCA_DESTAT_KPX_GMI[AddrV]=1. The lowest valid bit of the address is defined by MCA::KPX::GMI::MCA_DESTAT_KPX_GMI[AddrLsb].

_inst[GMICONTAINER[9:1,11,10:10,0]KPXGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2129

_ccd[11:0]_inst[GMICONTAINER[13:12]KPXGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_2139

_inst[GMICONTAINER[17:14]KPXGMI]_aliasSMN; GMICNTR[17:14]KPXx00000448; GMICNTR[17:14]KPX=1E[F:C]0_0000h

Bits	Description
63:0	ErrorAddr. Read-only. Reset: Cold,0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::KPX::GMI::MCA_DESTAT_KPX_GMI.

**MSRC001_041[2...3] [KPX_GMI Machine Check Control Mask]
(MCA::KPX::GMI::MCA_CTL_MASK_KPX_GMI)**

Read-write. Reset: 0000_0000_0000_0000h.

Inhibit detection of an error source.

`_inst[GMICONTAINER[9:1,11,10:10,0]KPXGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC001_0412``_ccd[11:0]_inst[GMICONTAINER[13:12]KPXGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC001_0413``_inst[GMICONTAINER[17:14]KPXGMI]_aliasSMN; GMICNTR[17:14]KPXx00000480; GMICNTR[17:14]KPX=1E[F:C]0_0000h`

Bits	Description
63:4	Reserved.
3	APB. Read-write. Reset: 0. PHY APB error
2	ARCData. Read-write. Reset: 0. ARC data buffer parity error
1	ARCIns. Read-write. Reset: 0. ARC instruction buffer parity error
0	RAMECC. Read-write. Reset: 0. RAM ECC Error.

**MSRC000_21[2...3]E [KPX_GMI Machine Check Syndrome Extended]
(MCA::KPX::GMI::MCA_SYND1_KPX_GMI)**

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::KPX::GMI::MCA_STATUS_KPX_GMI [Thread 0](#)`_inst[GMICONTAINER[9:1,11,10:10,0]KPXGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_212E``_ccd[11:0]_inst[GMICONTAINER[13:12]KPXGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_213E``_inst[GMICONTAINER[17:14]KPXGMI]_aliasSMN; GMICNTR[17:14]KPXx00000470; GMICNTR[17:14]KPX=1E[F:C]0_0000h`

Bits	Description
63:0	Syndrome. Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::KPX::GMI::MCA_SYND1_KPX_GMI register stores information associated with the error in MCA::KPX::GMI::MCA_STATUS_KPX_GMI or MCA_DESTAT. The register is meaningful if MCA::KPX::GMI::MCA_STATUS_KPX_GMI[SyndV]=1. When MCA::KPX::GMI::MCA_CONFIG_KPX_GMI[McaFruTextInMca]=1, MCA::KPX::GMI::MCA_SYND1_KPX_GMI stores ASCII FruText associated with the error.

**MSRC000_21[2...3]F [KPX_GMI Machine Check Syndrome Extended]
(MCA::KPX::GMI::MCA_SYND2_KPX_GMI)**

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::KPX::GMI::MCA_STATUS_KPX_GMI [Thread 0](#)`_inst[GMICONTAINER[9:1,11,10:10,0]KPXGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_212F``_ccd[11:0]_inst[GMICONTAINER[13:12]KPXGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_213F``_inst[GMICONTAINER[17:14]KPXGMI]_aliasSMN; GMICNTR[17:14]KPXx00000478; GMICNTR[17:14]KPX=1E[F:C]0_0000h`

Bits	Description
63:0	Syndrome. Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::KPX::GMI::MCA_SYND2_KPX_GMI register stores information associated with the error in MCA::KPX::GMI::MCA_STATUS_KPX_GMI or MCA_DESTAT. The register is meaningful if MCA::KPX::GMI::MCA_STATUS_KPX_GMI[SyndV]=1. When MCA::KPX::GMI::MCA_CONFIG_KPX_GMI[McaFruTextInMca]=1, MCA::KPX::GMI::MCA_SYND2_KPX_GMI stores ASCII FruText associated with the error.

3.2.5.18 PCS GMI

**MSR0000_0440...MSRC000_2110 [PCS_GMI Machine Check Control]
(MCA::PCS::GMI::MCA_CTL_PCS_GMI)**

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::PCS::GMI::MCA_CTL_PCS_GMI register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG_CTL. Does not affect error detection, correction, or logging.

_inst[GMICONTAINER[9:1,11,10:10,0]PCSGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSRLEGACY; MSR0000_0440

_ccd[11:0]_inst[GMICONTAINER[13:12]PCSGMI]_n[35:15,[13,11,9,7,5,3,1]]_aliasMSRLEGACY; MSR0000_0444

_inst[GMICONTAINER[9:1,11,10:10,0]PCSGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2100

_ccd[11:0]_inst[GMICONTAINER[13:12]PCSGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_2110

Bits	Description
63:32	Reserved.
31	TwixDataLoss. Read-write. Reset: 0. TwixDataLoss
30	LFDSFcinitTimeoutErr. Read-write. Reset: 0. LFDSFcinitTimeoutErr
29	LFDSTrainingTimeoutErr. Read-write. Reset: 0. LFDSTrainingTimeoutErr
28	RxCMDPktErr. Read-write. Reset: 0. RxCMDPktErr
27	LinkSubRxTimeoutErr. Read-write. Reset: 0. LinkSubRxTimeoutErr
26	LinkSubTxTimeoutErr. Read-write. Reset: 0. LinkSubTxTimeoutErr
25	RxLfdsFifoUnderflowErr. Read-write. Reset: 0. RxLfdsFifoUnderflowErr
24	RxLfdsFifoOverflowErr. Read-write. Reset: 0. RxLfdsFifoOverflowErr
23	TwixRxBuff. Read-write. Reset: 0. TwixRxBuff
22	DeskewAbortErr. Read-write. Reset: 0. DeskewAbortErr
21	RecoveryRelockAttemptErr. Read-write. Reset: 0. RecoveryRelockAttemptErr
20	RecoveryAttemptErr. Read-write. Reset: 0. RecoveryAttemptErr
19	ReadySerialAttemptErr. Read-write. Reset: 0. ReadySerialAttemptErr
18	ReadySerialTimeoutErr. Read-write. Reset: 0. ReadySerialTimeoutErr
17	RecoveryTimeoutErr. Read-write. Reset: 0. RecoveryTimeoutErr
16	FCInitTimeoutErr. Read-write. Reset: 0. FCInitTimeoutErr
15	DataStartupLimitErr. Read-write. Reset: 0. DataStartupLimitErr
14	TwixOffline. Read-write. Reset: 0. TwixOffline
13	DeskewErr. Read-write. Reset: 0. DeskewErr
12	ElasticFifoOverflowErr. Read-write. Reset: 0. ElasticFifoOverflowErr
11	ReplayFifoUnderflowErr. Read-write. Reset: 0. ReplayFifoUnderflowErr
10	ReplayFifoOverflowErr. Read-write. Reset: 0. ReplayFifoOverflowErr
9	TxTwixOverflow. Read-write. Reset: 0. TxTwixOverflow
8	ReplayBufParityErr. Read-write. Reset: 0. ReplayBufParityErr
7	TxTwixFifoUnderflow. Read-write. Reset: 0. TxTwixFifoUnderflow
6	BERExceededErr. Read-write. Reset: 0. BERExceededErr
5	CRCErr. Read-write. Reset: 0. CRCErr
4	RxFifoOverflowErr. Read-write. Reset: 0. RxFifoOverflowErr
3	RxFifoUnderflowErr. Read-write. Reset: 0. RxFifoUnderflowErr
2	ReplayParityTwix. Read-write. Reset: 0. ReplayParityTwix
1	TrainingErr. Read-write. Reset: 0. TrainingErr.
0	DataLossErr. Read-write. Reset: 0. DataLossErr

**MSR0000_0441...MSRC000_2111 [PCS_GMI Machine Check Status]
(MCA::PCS::GMI::MCA_STATUS_PCS_GMI)**

Reset: Cold,0000_0000_0000_0000h.

Logs information associated with errors.

_inst[GMICONTAINER[9:1,11,10:10,0]PCSGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSRLEGACY; MSR0000_0441

_ccd[11:0]_inst[GMICONTAINER[13:12]PCSGMI]_n[35:15,[13,11,9,7,5,3,1]]_aliasMSRLEGACY; MSR0000_0445

_inst[GMICONTAINER[9:1,11,10:10,0]PCSGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2101

_ccd[11:0]_inst[GMICONTAINER[13:12]PCSGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_2111

Bits	Description
63	Val. 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::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
62	Overflow. 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::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
61	UC. Reset: Cold,0. 1=The error was not corrected by hardware. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
60	En. Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::PCS::GMI::MCA_CTL_PCS_GMI. This bit is a copy of bit in MCA::PCS::GMI::MCA_CTL_PCS_GMI for this error. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	MiscV. Reset: Cold,0. 1=Valid thresholding in MCA::PCS::GMI::MCA_MISC0_PCS_GMI. 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::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
58	AddrV. Reset: Cold,0. 1=MCA::PCS::GMI::MCA_ADDR_PCS_GMI contains address information associated with the error. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	PCC. 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::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
56	ErrCoreIdVal. Reset: Cold,0. 1=The ErrCoreId field is valid. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
55	TCC. 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::PCS::GMI::MCA_STATUS_PCS_GMI[PCC]=0. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	RESERV54. Reset: Cold,0. MCA_STATUS Register Reserved bit. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
53	SyndV. Reset: Cold,0. 1=This error logged information in MCA::PCS::GMI::MCA_SYND_PCS_GMI. If MCA::PCS::GMI::MCA_SYND_PCS_GMI[ErrorPriority] is the same as the priority of the error in MCA::PCS::GMI::MCA_STATUS_PCS_GMI, then the information in MCA::PCS::GMI::MCA_SYND_PCS_GMI is associated with the error in MCA::PCS::GMI::MCA_STATUS_PCS_GMI. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	RESERV47. 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	CECC . 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	UECC . 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	Deferred . 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	Poison . 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	RESERV41 . 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	Scrub . 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	RESERV38 . 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	ErrCoreId . 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	RESERV30 . 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	AddrLsb . Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::PCS::GMI::MCA_ADDR_PCS_GMI[ErrorAddr]. A value of 0 indicates that MCA::PCS::GMI::MCA_ADDR_PCS_GMI[63:0] contains a valid byte address. A value of 6 indicates that MCA::PCS::GMI::MCA_ADDR_PCS_GMI[63:6] contains a valid cache line address and that MCA::PCS::GMI::MCA_ADDR_PCS_GMI[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::PCS::GMI::MCA_ADDR_PCS_GMI[63:12] contain a valid 4KB memory page and that MCA::PCS::GMI::MCA_ADDR_PCS_GMI[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	RESERV22 . 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	ErrorCodeExt . 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::PCS::GMI::MCA_CTL_PCS_GMI 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	ErrorCode . 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 103: MCA_STATUS_PCS_GMI

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
DataLossErr	0x0	0	0	0	0	0	0
TrainingErr	0x1	0	0	0	0	0	0
ReplayParity Twix	0x2	0	0	0	0	0	0

RxFifoUnderflowErr	0x3	0	0	0	0	0	0
RxFifoOverflowErr	0x4	0	0	0	0	0	0
CRCErr	0x5	0	0	0	0	0	0
BERExceededErr	0x6	0	0	0	0	0	0
TxTwixFifoUnderflow	0x7	0	0	0	0	0	0
ReplayBufParityErr	0x8	0	0	0	0	0	0
TxTwixOverflow	0x9	0	0	0	0	0	0
ReplayFifoOverflowErr	0xa	0	0	0	0	0	0
ReplayFifoUnderflowErr	0xb	0	0	0	0	0	0
ElasticFifoOverflowErr	0xc	0	0	0	0	0	0
DeskewErr	0xd	0	0	0	0	0	0
TwixOffline	0xe	0	0	0	0	0	0
DataStartupLimitErr	0xf	0	0	0	0	0	0
FCInitTimeoutErr	0x10	0	0	0	0	0	0
RecoveryTimeoutErr	0x11	0	0	0	0	0	0
ReadySerialTimeoutErr	0x12	0	0	0	0	0	0
ReadySerialAttemptErr	0x13	0	0	0	0	0	0
RecoveryAttemptErr	0x14	0	0	0	0	0	0
RecoveryRelockAttemptErr	0x15	0	0	0	0	0	0
DeskewAbortErr	0x16	0	0	0	0	0	0
TwixRxBuff	0x17	0	0	0	0	0	0
RxLfdsFifoOverflowErr	0x18	0	0	0	0	0	0
RxLfdsFifoUnderflowErr	0x19	0	0	0	0	0	0
LinkSubTxTimeoutErr	0x1a	0	0	0	0	0	0
LinkSubRxTimeoutErr	0x1b	0	0	0	0	0	0
RxCMDPktErr	0x1c	0	0	0	0	0	0

LFDSTrainin gTimeoutErr	0x1d	0	0	0	0	0	0
LFDSFcinitT imeoutErr	0x1e	0	0	0	0	0	0
TwixDataLo ss	0x1f	0	0	0	0	0	0

**MSR0000_0442...MSRC000_2112 [PCS_GMI Machine Check Address]
(MCA::PCS::GMI::MCA_ADDR_PCS_GMI)**

Read-only. Reset: Cold,0000_0000_0000_0000h.

MCA::PCS::GMI::MCA_ADDR_PCS_GMI stores an address and other information associated with the error in MCA::PCS::GMI::MCA_STATUS_PCS_GMI. The register is only meaningful if MCA::PCS::GMI::MCA_STATUS_PCS_GMI[Val]=1 and MCA::PCS::GMI::MCA_STATUS_PCS_GMI[AddrV]=1.

_inst[GMICONTAINER[9:1,11,10:10,0]PCSGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSRLEGACY; MSR0000_0442

_ccd[11:0]_inst[GMICONTAINER[13:12]PCSGMI]_n[35:15,[13,11,9,7,5,3,1]]_aliasMSRLEGACY; MSR0000_0446

_inst[GMICONTAINER[9:1,11,10:10,0]PCSGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2102

_ccd[11:0]_inst[GMICONTAINER[13:12]PCSGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_2112

Bits	Description
63:0	ErrorAddr. Read-only. Reset: Cold,0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::PCS::GMI::MCA_STATUS_PCS_GMI.

Table 104: MCA_ADDR_PCS_GMI

Error Type	Bits	Description
DataLossErr	[63:0]	Reserved
TrainingErr	[63:0]	Reserved
ReplayParityTwix	[63:0]	Reserved
RxFifoUnderflowErr	[63:0]	Reserved
RxFifoOverflowErr	[63:0]	Reserved
CRCErr	[63:0]	Reserved
BERExceededErr	[63:0]	Reserved
TxTwixFifoUnderflow	[63:0]	Reserved
ReplayBufParityErr	[63:0]	Reserved
TxTwixOverflow	[63:0]	Reserved
ReplayFifoOverflowErr	[63:0]	Reserved
ReplayFifoUnderflowErr	[63:0]	Reserved
ElasticFifoOverflowErr	[63:0]	Reserved
DeskewErr	[63:0]	Reserved
TwixOffline	[63:0]	Reserved
DataStartupLimitErr	[63:0]	Reserved
FCInitTimeoutErr	[63:0]	Reserved
RecoveryTimeoutErr	[63:0]	Reserved
ReadySerialTimeoutErr	[63:0]	Reserved
ReadySerialAttemptErr	[63:0]	Reserved
RecoveryAttemptErr	[63:0]	Reserved
RecoveryRelockAttemptErr	[63:0]	Reserved
DeskewAbortErr	[63:0]	Reserved
TwixRxBuff	[63:0]	Reserved
RxLfdsFifoOverflowErr	[63:0]	Reserved
RxLfdsFifoUnderflowErr	[63:0]	Reserved
LinkSubTxTimeoutErr	[63:0]	Reserved

LinkSubRxTimeoutErr	[63:0]	Reserved
RxCMDPktErr	[63:0]	Reserved
LFDSTrainingTimeoutErr	[63:0]	Reserved
LFDSFcinitTimeoutErr	[63:0]	Reserved
TwixDataLoss	[63:0]	Reserved

**MSR0000_0443...MSRC000_2113 [PCS_GMI Machine Check Miscellaneous 0]
(MCA::PCS::GMI::MCA_MISC0_PCS_GMI)**

Log miscellaneous information associated with errors.

_inst[GMICONTAINER[9:1,11,10:10,0]PCSGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSRLEGACY; MSR0000_0443

_ccd[11:0]_inst[GMICONTAINER[13:12]PCSGMI]_n[35:15,[13,11,9,7,5,3,1]]_aliasMSRLEGACY; MSR0000_0447

_inst[GMICONTAINER[9:1,11,10:10,0]PCSGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2103

_ccd[11:0]_inst[GMICONTAINER[13:12]PCSGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_2113

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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::PCS::GMI::MCA_MISC0_PCS_GMI[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	LvtOffset. 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::PCS::GMI::MCA_MISC0_PCS_GMI[Locked]) ? Read-write : Read-only.
51	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn] !MCA::PCS::GMI::MCA_MISC0_PCS_GMI[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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::PCS::GMI::MCA_MISC0_PCS_GMI[Locked]) ? Read-write : Read-only.
48	Ovrflw. 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::PCS::GMI::MCA_MISC0_PCS_GMI[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	ErrCnt. 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::PCS::GMI::MCA_MISC0_PCS_GMI[Locked]) ? Read-write : Read-only.
31:24	BlkPtr. Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

**MSRC000_21[0...1]4 [PCS_GMI Machine Check Configuration]
(MCA::PCS::GMI::MCA_CONFIG_PCS_GMI)**

Reset: 0000_0002_0000_0125h.

Controls configuration of the associated machine check bank.

_inst[GMICONTAINER[9:1,11,10:10,0]PCSGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2104

_ccd[11:0]_inst[GMICONTAINER[13:12]PCSGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_2114

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	LogDeferredInMcaStat. Read-write. Reset: 0. Init: BIOS,1. 1=Log deferred errors in MCA::PCS::GMI::MCA_STATUS_PCS_GMI and MCA::PCS::GMI::MCA_ADDR_PCS_GMI in addition to MCA::PCS::GMI::MCA_DESTAT_PCS_GMI and MCA::PCS::GMI::MCA_DEADDR_PCS_GMI. 0=Only log deferred errors in MCA::PCS::GMI::MCA_DESTAT_PCS_GMI and MCA::PCS::GMI::MCA_DEADDR_PCS_GMI. This bit does not affect logging of deferred errors in MCA::PCS::GMI::MCA_SYND_PCS_GMI, MCA::PCS::GMI::MCA_MISC0_PCS_GMI.
33	Reserved.
32	McaXEnable. 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	IntPresent. Read-only, Volatile . Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::PCS::GMI::MCA_CONFIG_PCS_GMI[IntEn].
9	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::PCS::GMI::MCA_CONFIG_PCS_GMI[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. Read-only. Reset: 1. 1=MCA::PCS::GMI::MCA_CONFIG_PCS_GMI[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::PCS::GMI::MCA_CONFIG_PCS_GMI[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	DeferredErrorLoggingSupported. Read-only. Reset: 1. 1=Deferred errors are supported in this MCA bank, and MCA::PCS::GMI::MCA_CONFIG_PCS_GMI[LogDeferredInMcaStat] controls the logging behavior of these errors. MCA::PCS::GMI::MCA_DESTAT_PCS_GMI and MCA::PCS::GMI::MCA_DEADDR_PCS_GMI are supported in this MCA bank. 0=Deferred errors are not supported in this bank.
1	Reserved.
0	McaX. Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::PCS::GMI::MCA_MISC0_PCS_GMI[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::PCS::GMI::MCA_STATUS_PCS_GMI[TCC] is present.

MSRC000_21[0...1]5 [PCS_GMI IP Identification] (MCA::PCS::GMI::MCA_IPID_PCS_GMI)

Reset: 0000_0241_0000_0000h.

The MCA::PCS::GMI::MCA_IPID_PCS_GMI register is used by software to determine what IP type and revision is associated with the MCA bank.

_inst[GMICONTAINER[9:1,11,10:10,0]PCSGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2105

_ccd[11:0]_inst[GMICONTAINER[13:12]PCSGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_2115

Bits	Description
63:48	McaType . Read-only. Reset: 0000h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi . 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	HardwareID . Read-only. Reset: 241h. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _ccd0_instGMICONTAINER12PCSGMI_n1_aliasMSR: 3070_9200h
	Init: _ccd0_instGMICONTAINER13PCSGMI_n24_aliasMSR: 3080_9200h
	Init: _ccd1_instGMICONTAINER12PCSGMI_n3_aliasMSR: 3270_9200h
	Init: _ccd1_instGMICONTAINER13PCSGMI_n25_aliasMSR: 3280_9200h
	Init: _ccd2_instGMICONTAINER12PCSGMI_n5_aliasMSR: 3470_9200h
	Init: _ccd2_instGMICONTAINER13PCSGMI_n26_aliasMSR: 3480_9200h
	Init: _ccd3_instGMICONTAINER12PCSGMI_n7_aliasMSR: 3670_9200h
	Init: _ccd3_instGMICONTAINER13PCSGMI_n27_aliasMSR: 3680_9200h
	Init: _ccd4_instGMICONTAINER12PCSGMI_n9_aliasMSR: 3870_9200h
	Init: _ccd4_instGMICONTAINER13PCSGMI_n28_aliasMSR: 3880_9200h
	Init: _ccd5_instGMICONTAINER12PCSGMI_n11_aliasMSR: 3A70_9200h
	Init: _ccd5_instGMICONTAINER13PCSGMI_n29_aliasMSR: 3A80_9200h
	Init: _ccd6_instGMICONTAINER12PCSGMI_n13_aliasMSR: 3C70_9200h
	Init: _ccd6_instGMICONTAINER13PCSGMI_n30_aliasMSR: 3C80_9200h
	Init: _ccd7_instGMICONTAINER12PCSGMI_n15_aliasMSR: 3E70_9200h
	Init: _ccd7_instGMICONTAINER13PCSGMI_n31_aliasMSR: 3E80_9200h
	Init: _ccd8_instGMICONTAINER12PCSGMI_n17_aliasMSR: 4A70_9200h
	Init: _ccd8_instGMICONTAINER13PCSGMI_n32_aliasMSR: 4A80_9200h
	Init: _ccd9_instGMICONTAINER12PCSGMI_n19_aliasMSR: 4C70_9200h
	Init: _ccd9_instGMICONTAINER13PCSGMI_n33_aliasMSR: 4C80_9200h
	Init: _ccd10_instGMICONTAINER12PCSGMI_n21_aliasMSR: 4E70_9200h
	Init: _ccd10_instGMICONTAINER13PCSGMI_n34_aliasMSR: 4E80_9200h
	Init: _ccd11_instGMICONTAINER12PCSGMI_n23_aliasMSR: 5070_9200h
	Init: _ccd11_instGMICONTAINER13PCSGMI_n35_aliasMSR: 5080_9200h
	Init: _instGMICONTAINER0PCSGMI_n0_aliasMSR: 1330_9200h
	Init: _instGMICONTAINER10PCSGMI_n20_aliasMSR: 19A0_9200h
	Init: _instGMICONTAINER11PCSGMI_n22_aliasMSR: 19B0_9200h
	Init: _instGMICONTAINER1PCSGMI_n2_aliasMSR: 1340_9200h
	Init: _instGMICONTAINER2PCSGMI_n4_aliasMSR: 1350_9200h
	Init: _instGMICONTAINER3PCSGMI_n6_aliasMSR: 1360_9200h
	Init: _instGMICONTAINER4PCSGMI_n8_aliasMSR: 1370_9200h
	Init: _instGMICONTAINER5PCSGMI_n10_aliasMSR: 1380_9200h
	Init: _instGMICONTAINER6PCSGMI_n12_aliasMSR: 1390_9200h
	Init: _instGMICONTAINER7PCSGMI_n14_aliasMSR: 17C0_9200h
	Init: _instGMICONTAINER8PCSGMI_n16_aliasMSR: 17D0_9200h
	Init: _instGMICONTAINER9PCSGMI_n18_aliasMSR: 17E0_9200h

MSRC000_21[0...1]6 [PCS_GMI Machine Check Syndrome] (MCA::PCS::GMI::MCA_SYND_PCS_GMI)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::PCS::GMI::MCA_STATUS_PCS_GMI [Thread 0](#)

_inst[GMICONTAINER[9:1,11,10:10,0]PCSGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2106

_ccd[11:0]_inst[GMICONTAINER[13:12]PCSGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_2116

Bits	Description
63:32	Syndrome. Read-write, Volatile . Reset: Cold, 0000_0000h. Contains the syndrome, if any, associated with the error logged in MCA::PCS::GMI::MCA_STATUS_PCS_GMI. The low-order bit of the syndrome is stored in bit 0, and the syndrome has a length specified by MCA::PCS::GMI::MCA_SYND_PCS_GMI[Length]. The Syndrome field is only valid when MCA::PCS::GMI::MCA_SYND_PCS_GMI[Length] is not 0.
31:27	Reserved.
26:24	ErrorPriority. Read-write, Volatile . Reset: Cold, 0h. Encodes the priority of the error logged in MCA::PCS::GMI::MCA_SYND_PCS_GMI. 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	Length. Read-write, Volatile . Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::PCS::GMI::MCA_SYND_PCS_GMI[Syndrome]. Length values greater than 32 (decimal) are interpreted as equal to 32 (decimal). A value of 0 indicates that there is no valid syndrome in MCA::PCS::GMI::MCA_SYND_PCS_GMI. For example, a syndrome length of 9 means that MCA::PCS::GMI::MCA_SYND_PCS_GMI[Syndrome] bits [8:0] contains a valid syndrome.
17:0	ErrorInformation. Read-write, Volatile . Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 105 [MCA_SYND_PCS_GMI].

Table 105: MCA_SYND_PCS_GMI

Error Type	Bits	Description
DataLossErr	[17:0]	Reserved
TrainingErr	[17:0]	Reserved
ReplayParityTwix	[17:0]	Reserved
RxFifoUnderflowErr	[17:0]	Reserved
RxFifoOverflowErr	[17:0]	Reserved
CRCErr	[17:0]	Reserved
BERExceededErr	[17:0]	Reserved
TxTwixFifoUnderflow	[17:0]	Reserved
ReplayBufParityErr	[17:0]	Reserved
TxTwixOverflow	[17:0]	Reserved
ReplayFifoOverflowErr	[17:0]	Reserved
ReplayFifoUnderflowErr	[17:0]	Reserved
ElasticFifoOverflowErr	[17:0]	Reserved
DeskewErr	[17:0]	Reserved
TwixOffline	[17:0]	Reserved
DataStartupLimitErr	[17:0]	Reserved
FCInitTimeoutErr	[17:0]	Reserved
RecoveryTimeoutErr	[17:0]	Reserved
ReadySerialTimeoutErr	[17:0]	Reserved
ReadySerialAttemptErr	[17:0]	Reserved
RecoveryAttemptErr	[17:0]	Reserved
RecoveryRelockAttemptErr	[17:0]	Reserved
DeskewAbortErr	[17:0]	Reserved
TwixRxBuff	[17:0]	Reserved
RxLfdsFifoOverflowErr	[17:0]	Reserved

RxLdsFifoUnderflowErr	[17:0]	Reserved
LinkSubTxTimeoutErr	[17:0]	Reserved
LinkSubRxTimeoutErr	[17:0]	Reserved
RxCMDPktErr	[17:0]	Reserved
LFDSTrainingTimeoutErr	[17:0]	Reserved
LFDSFcinitTimeoutErr	[17:0]	Reserved
TwixDataLoss	[17:0]	Reserved

MSRC000_21[0...1]8 [PCS_GMI Machine Check Deferred Error Status] (MCA::PCS::GMI::MCA_DESTAT_PCS_GMI)

Reset: Cold,0000_0000_0000_0000h.

Holds status information for the first deferred error seen in this bank.

_inst[GMICONTAINER[9:1,11,10:10,0]PCSGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2108

_ccd[11:0]_inst[GMICONTAINER[13:12]PCSGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_2118

Bits	Description
63	Val. Read-write, Volatile . Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	Overflow. 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 overwrite priorities.)
61:59	RESERV4. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	AddrV. Read-write, Volatile . Reset: Cold,0. 1=MCA::PCS::GMI::MCA_DEADDR_PCS_GMI contains address information associated with the error.
57:54	RESERV3. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	SyndV. Read-write, Volatile . Reset: Cold,0. 1=This error logged information in MCA::PCS::GMI::MCA_SYND_PCS_GMI. If MCA::PCS::GMI::MCA_SYND_PCS_GMI[ErrorPriority] is the same as the priority of the error in MCA::PCS::GMI::MCA_STATUS_PCS_GMI, then the information in MCA::PCS::GMI::MCA_SYND_PCS_GMI is associated with the error in MCA::PCS::GMI::MCA_DESTAT_PCS_GMI.
52:45	RESERV2. Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	Deferred. 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	RESERV1. Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	AddrLsb. Read-write, Volatile . Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::PCS::GMI::MCA_ADDR_PCS_GMI[ErrorAddr]. A value of 0 indicates that MCA::PCS::GMI::MCA_ADDR_PCS_GMI[63:0] contains a valid byte address. A value of 6 indicates that MCA::PCS::GMI::MCA_ADDR_PCS_GMI[63:6] contains a valid cache line address and that MCA::PCS::GMI::MCA_ADDR_PCS_GMI[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::PCS::GMI::MCA_ADDR_PCS_GMI[63:12] contain a valid 4KB memory page and that MCA::PCS::GMI::MCA_ADDR_PCS_GMI[11:0] should be ignored by error handling software.
23:22	RESERV0. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	ErrorCodeExt. 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	ErrorCode. Read-write, Volatile . Reset: Cold,0000h. Error code for this error.

MSRC000_21[0...1]9 [PCS_GMI Deferred Error Address] (MCA::PCS::GMI::MCA_DEADDR_PCS_GMI)

Read-only. Reset: Cold,0000_0000_0000_0000h.

The MCA::PCS::GMI::MCA_DEADDR_PCS_GMI register stores the address associated with the error in MCA::PCS::GMI::MCA_DESTAT_PCS_GMI. The register is only meaningful if MCA::PCS::GMI::MCA_DESTAT_PCS_GMI[Val]=1 and MCA::PCS::GMI::MCA_DESTAT_PCS_GMI[AddrV]=1. The lowest valid bit of the address is defined by MCA::PCS::GMI::MCA_DESTAT_PCS_GMI[AddrLsb].

_inst[GMICONTAINER[9:1,11,10:10,0]PCSGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_2109

_ccd[11:0]_inst[GMICONTAINER[13:12]PCSGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_2119

Bits	Description
63:0	ErrorAddr. Read-only. Reset: Cold,0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::PCS::GMI::MCA_DESTAT_PCS_GMI.

**MSRC001_041[0...1] [PCS_GMI Machine Check Control Mask]
(MCA::PCS::GMI::MCA_CTL_MASK_PCS_GMI)**

Read-write. Reset: 0000_0000_0000_0000h.

Inhibit detection of an error source.

_inst[GMICONTAINER[9:1,11,10:10,0]PCSGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC001_0410

_ccd[11:0]_inst[GMICONTAINER[13:12]PCSGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC001_0411

Bits	Description
63:32	Reserved.
31	TwixDataLoss. Read-write. Reset: 0. TwixDataLoss
30	LFDSFcinitTimeoutErr. Read-write. Reset: 0. LFDSFcinitTimeoutErr
29	LFDSTrainingTimeoutErr. Read-write. Reset: 0. LFDSTrainingTimeoutErr
28	RxCMDPktErr. Read-write. Reset: 0. RxCMDPktErr
27	LinkSubRxTimeoutErr. Read-write. Reset: 0. LinkSubRxTimeoutErr
26	LinkSubTxTimeoutErr. Read-write. Reset: 0. LinkSubTxTimeoutErr
25	RxLfdsFifoUnderflowErr. Read-write. Reset: 0. RxLfdsFifoUnderflowErr
24	RxLfdsFifoOverflowErr. Read-write. Reset: 0. RxLfdsFifoOverflowErr
23	TwixRxBuff. Read-write. Reset: 0. TwixRxBuff
22	DeskewAbortErr. Read-write. Reset: 0. DeskewAbortErr
21	RecoveryRelockAttemptErr. Read-write. Reset: 0. RecoveryRelockAttemptErr
20	RecoveryAttemptErr. Read-write. Reset: 0. RecoveryAttemptErr
19	ReadySerialAttemptErr. Read-write. Reset: 0. ReadySerialAttemptErr
18	ReadySerialTimeoutErr. Read-write. Reset: 0. ReadySerialTimeoutErr
17	RecoveryTimeoutErr. Read-write. Reset: 0. RecoveryTimeoutErr
16	FCInitTimeoutErr. Read-write. Reset: 0. FCInitTimeoutErr
15	DataStartupLimitErr. Read-write. Reset: 0. DataStartupLimitErr
14	TwixOffline. Read-write. Reset: 0. TwixOffline
13	DeskewErr. Read-write. Reset: 0. DeskewErr
12	ElasticFifoOverflowErr. Read-write. Reset: 0. ElasticFifoOverflowErr
11	ReplayFifoUnderflowErr. Read-write. Reset: 0. ReplayFifoUnderflowErr
10	ReplayFifoOverflowErr. Read-write. Reset: 0. ReplayFifoOverflowErr
9	TxTwixOverflow. Read-write. Reset: 0. TxTwixOverflow
8	ReplayBufParityErr. Read-write. Reset: 0. ReplayBufParityErr
7	TxTwixFifoUnderflow. Read-write. Reset: 0. TxTwixFifoUnderflow
6	BERExceededErr. Read-write. Reset: 0. BERExceededErr
5	CRCErr. Read-write. Reset: 0. CRCErr
4	RxFifoOverflowErr. Read-write. Reset: 0. RxFifoOverflowErr
3	RxFifoUnderflowErr. Read-write. Reset: 0. RxFifoUnderflowErr
2	ReplayParityTwix. Read-write. Reset: 0. ReplayParityTwix
1	TrainingErr. Read-write. Reset: 0. TrainingErr.
0	DataLossErr. Read-write. Reset: 0. DataLossErr

MSRC000_21[0...1]E [PCS_GMI Machine Check Syndrome Extended] (MCA::PCS::GMI::MCA_SYND1_PCS_GMI)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::PCS::GMI::MCA_STATUS_PCS_GMI [Thread 0](#)

_inst[GMICONTAINER[9:1,11,10:10,0]PCSGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_210E

_ccd[11:0]_inst[GMICONTAINER[13:12]PCSGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_211E

Bits	Description
63:0	Syndrome. Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::PCS::GMI::MCA_SYND1_PCS_GMI register stores information associated with the error in MCA::PCS::GMI::MCA_STATUS_PCS_GMI or MCA_DESTAT. The register is meaningful if MCA::PCS::GMI::MCA_STATUS_PCS_GMI[SyndV]=1. When MCA::PCS::GMI::MCA_CONFIG_PCS_GMI[McaFruTextInMca]=1, MCA::PCS::GMI::MCA_SYND1_PCS_GMI stores ASCII FruText associated with the error.

MSRC000_21[0...1]F [PCS_GMI Machine Check Syndrome Extended] (MCA::PCS::GMI::MCA_SYND2_PCS_GMI)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::PCS::GMI::MCA_STATUS_PCS_GMI [Thread 0](#)

_inst[GMICONTAINER[9:1,11,10:10,0]PCSGMI]_n[22,20,18,16,14,12,10,8,6,4,2,0]_aliasMSR; MSRC000_210F

_ccd[11:0]_inst[GMICONTAINER[13:12]PCSGMI]_n[35:23,[21,19,17,15,13,11,9,7,5,3,1]]_aliasMSR; MSRC000_211F

Bits	Description
63:0	Syndrome. Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::PCS::GMI::MCA_SYND2_PCS_GMI register stores information associated with the error in MCA::PCS::GMI::MCA_STATUS_PCS_GMI or MCA_DESTAT. The register is meaningful if MCA::PCS::GMI::MCA_STATUS_PCS_GMI[SyndV]=1. When MCA::PCS::GMI::MCA_CONFIG_PCS_GMI[McaFruTextInMca]=1, MCA::PCS::GMI::MCA_SYND2_PCS_GMI stores ASCII FruText associated with the error.

3.2.5.19 PCS XGMI

**MSR0000_0478...MSRC000_21E0 [PCS_XGMI Machine Check Control]
(MCA::PCS::XGMI::MCA_CTL_PCS_XGMI)**

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::PCS::XGMI::MCA_CTL_PCS_XGMI register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG_CTL. Does not affect error detection, correction, or logging.

_inst[SERDESBP[3,1]PCSXGMI,SERDESBG[2:0]PCSXGMI,SERDESAG3PCSXGMI]_n[5:0]_aliasMSRLEGACY; MSR0000_0478

_inst[SERDESBP[3,1]PCSXGMI,SERDESBG[2:0]PCSXGMI,SERDESAG3PCSXGMI]_n[5:0]_aliasMSR; MSRC000_21E0

Bits	Description
63:29	Reserved.
28	RxCMDPktErr. Read-write. Reset: 0. RxCMDPktErr for xGMI3
27	LinkSubRxTimeoutErr. Read-write. Reset: 0. LinkSubRxTimeoutErr for xGMI3
26	LinkSubTxTimeoutErr. Read-write. Reset: 0. LinkSubTxTimeoutErr for xGMI3
25	RxReplayTimeoutErr. Read-write. Reset: 0. RxReplayTimeoutErr for xGMI3
24	TxReplayTimeoutErr. Read-write. Reset: 0. TxReplayTimeoutErr for xGMI3
23	SyncHdrErr. Read-write. Reset: 0. SyncHdrErr for xGMI3
22	ReplayAttemptErr. Read-write. Reset: 0. ReplayAttemptErr for xGMI3
21	RecoveryRelockAttemptErr. Read-write. Reset: 0. RecoveryRelockAttemptErr
20	RecoveryAttemptErr. Read-write. Reset: 0. RecoveryAttemptErr
19	ReadySerialAttemptErr. Read-write. Reset: 0. ReadySerialAttemptErr
18	ReadySerialTimeoutErr. Read-write. Reset: 0. ReadySerialTimeoutErr
17	RecoveryTimeoutErr. Read-write. Reset: 0. RecoveryTimeoutErr
16	FCInitTimeoutErr. Read-write. Reset: 0. FCInitTimeoutErr
15	DataStartupLimitErr. Read-write. Reset: 0. DataStartupLimitErr
14	FlowCtrlCRCErr. Read-write. Reset: 0. FlowCtrlCRCErr for xGMI3
13	DeskewErr. Read-write. Reset: 0. DeskewErr
12	ElasticFifoOverflowErr. Read-write. Reset: 0. ElasticFifoOverflowErr
11	ReplayFifoUnderflowErr. Read-write. Reset: 0. ReplayFifoUnderflowErr
10	ReplayFifoOverflowErr. Read-write. Reset: 0. ReplayFifoOverflowErr
9	DataParityErr. Read-write. Reset: 0. DataParityErr
8	ReplayBufParityErr. Read-write. Reset: 0. ReplayBufParityErr
7	TxMetaDataErr_TxVcidDataErr. Read-write. Reset: 0. TxVcidDataErr for xGMI3.
6	BERExceededErr. Read-write. Reset: 0. BERExceededErr
5	CRCErr. Read-write. Reset: 0. CRCErr
4	RxFifoOverflowErr. Read-write. Reset: 0. RxFifoOverflowErr for xGMI3
3	RxFifoUnderflowErr. Read-write. Reset: 0. RxFifoUnderflowErr for xGMI3
2	FlowCtrlAckErr. Read-write. Reset: 0. FlowCtrlAckErr for xGMI3.
1	TrainingErr. Read-write. Reset: 0. TrainingErr.
0	DataLossErr. Read-write. Reset: 0. DataLossErr

**MSR0000_0479...MSRC000_21E1 [PCS_XGMI Machine Check Status]
(MCA::PCS::XGMI::MCA_STATUS_PCS_XGMI)**

Reset: Cold,0000_0000_0000_0000h.

Logs information associated with errors.

_inst[SERDESBP[3,1]PCSXGMI,SERDESBG[2:0]PCSXGMI,SERDESAG3PCSXGMI]_n[5:0]_aliasMSRLEGACY; MSR0000_0479

_inst[SERDESBP[3,1]PCSXGMI,SERDESBG[2:0]PCSXGMI,SERDESAG3PCSXGMI]_n[5:0]_aliasMSR; MSRC000_21E1

Bits	Description
63	Val. 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::Msrr: HWCR[McStatusWrEn] ? Read-write : Read, Write-0-only , Error-on-write-1 .
62	Overflow. 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::Msrr: HWCR[McStatusWrEn] ? Read-write : Read, Write-0-only , Error-on-write-1 .
61	UC. Reset: Cold,0. 1=The error was not corrected by hardware. AccessType: Core::X86::Msrr: HWCR[McStatusWrEn] ? Read-write : Read, Write-0-only , Error-on-write-1 .
60	En. Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::PCS::XGMI::MCA_CTL_PCS_XGMI. This bit is a copy of bit in MCA::PCS::XGMI::MCA_CTL_PCS_XGMI for this error. AccessType: Core::X86::Msrr: HWCR[McStatusWrEn] ? Read-write : Read, Write-0-only , Error-on-write-1 .
59	MiscV. Reset: Cold,0. 1=Valid thresholding in MCA::PCS::XGMI::MCA_MISC0_PCS_XGMI. 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::Msrr: HWCR[McStatusWrEn] ? Read-write : Read, Write-0-only , Error-on-write-1 .
58	AddrV. Reset: Cold,0. 1=MCA::PCS::XGMI::MCA_ADDR_PCS_XGMI contains address information associated with the error. AccessType: Core::X86::Msrr: HWCR[McStatusWrEn] ? Read-write : Read, Write-0-only , Error-on-write-1 .
57	PCC. 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::Msrr: HWCR[McStatusWrEn] ? Read-write : Read, Write-0-only , Error-on-write-1 .
56	ErrCoreIdVal. Reset: Cold,0. 1=The ErrCoreId field is valid. AccessType: Core::X86::Msrr: HWCR[McStatusWrEn] ? Read-write : Read, Write-0-only , Error-on-write-1 .
55	TCC. 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::PCS::XGMI::MCA_STATUS_PCS_XGMI[PCC]=0. AccessType: Core::X86::Msrr: HWCR[McStatusWrEn] ? Read-write : Read, Write-0-only , Error-on-write-1 .
54	RESERV54. Reset: Cold,0. MCA_STATUS Register Reserved bit. AccessType: Core::X86::Msrr: HWCR[McStatusWrEn] ? Read-write : Read, Write-0-only , Error-on-write-1 .
53	SyndV. Reset: Cold,0. 1=This error logged information in MCA::PCS::XGMI::MCA_SYND_PCS_XGMI. If MCA::PCS::XGMI::MCA_SYND_PCS_XGMI[ErrorPriority] is the same as the priority of the error in MCA::PCS::XGMI::MCA_STATUS_PCS_XGMI, then the information in MCA::PCS::XGMI::MCA_SYND_PCS_XGMI is associated with the error in MCA::PCS::XGMI::MCA_STATUS_PCS_XGMI. AccessType: Core::X86::Msrr: HWCR[McStatusWrEn] ? Read-write : Read, Write-0-only , Error-on-write-1 .
52	Reserved.
51:47	RESERV47. 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	CECC. 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	UECC . 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	Deferred . 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	Poison . 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	RESERV41 . 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	Scrub . 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	RESERV38 . 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	ErrCoreId . 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	RESERV30 . 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	AddrLsb . Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::PCS::XGMI::MCA_ADDR_PCS_XGMI[ErrorAddr]. A value of 0 indicates that MCA::PCS::XGMI::MCA_ADDR_PCS_XGMI[63:0] contains a valid byte address. A value of 6 indicates that MCA::PCS::XGMI::MCA_ADDR_PCS_XGMI[63:6] contains a valid cache line address and that MCA::PCS::XGMI::MCA_ADDR_PCS_XGMI[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::PCS::XGMI::MCA_ADDR_PCS_XGMI[63:12] contain a valid 4KB memory page and that MCA::PCS::XGMI::MCA_ADDR_PCS_XGMI[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	RESERV22 . 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	ErrorCodeExt . 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::PCS::XGMI::MCA_CTL_PCS_XGMI 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	ErrorCode . 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 106: MCA_STATUS_PCS_XGMI

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
DataLossErr	0x0	0	0	0	0	0	0
TrainingErr	0x1	0	0	0	0	0	0
FlowCtrlAck Err	0x2	0	0	0	0	0	0
RxFifoUnderflowErr	0x3	0	0	0	0	0	0

RxFifoOverflowErr	0x4	0	0	0	0	0	0
CRCErr	0x5	0	0	0	0	0	0
BERExceededErr	0x6	0	0	0	0	0	0
TxMetadataErr_TxVcidDataErr	0x7	0	0	0	0	0	0
ReplayBufParityErr	0x8	0	0	0	0	0	0
DataParityErr	0x9	0	0	0	0	0	0
ReplayFifoOverflowErr	0xa	0	0	0	0	0	0
ReplayFifoUnderflowErr	0xb	0	0	0	0	0	0
ElasticFifoOverflowErr	0xc	0	0	0	0	0	0
DeskewErr	0xd	0	0	0	0	0	0
FlowCtrlCRCErr	0xe	0	0	0	0	0	0
DataStartupLimitErr	0xf	0	0	0	0	0	0
FCInitTimeoutErr	0x10	0	0	0	0	0	0
RecoveryTimeoutErr	0x11	0	0	0	0	0	0
ReadySerialTimeoutErr	0x12	0	0	0	0	0	0
ReadySerialAttemptErr	0x13	0	0	0	0	0	0
RecoveryAttemptErr	0x14	0	0	0	0	0	0
RecoveryRelockAttemptErr	0x15	0	0	0	0	0	0
ReplayAttemptErr	0x16	0	0	0	0	0	0
SyncHdrErr	0x17	0	0	0	0	0	0
TxReplayTimeoutErr	0x18	0	0	0	0	0	0
RxReplayTimeoutErr	0x19	0	0	0	0	0	0
LinkSubTxTimeoutErr	0x1a	0	0	0	0	0	0
LinkSubRxTimeoutErr	0x1b	0	0	0	0	0	0
RxCMDPktErr	0x1c	0	0	0	0	0	0
Reserved_29	0x1d	0	0	0	0	0	0

Reserved_30	0x1e	0	0	0	0	0	0
Reserved_31	0x1f	0	0	0	0	0	0

**MSR0000_047A...MSRC000_21E2 [PCS_XGMI Machine Check Address]
(MCA::PCS::XGMI::MCA_ADDR_PCS_XGMI)**

Read-only. Reset: Cold,0000_0000_0000_0000h.

MCA::PCS::XGMI::MCA_ADDR_PCS_XGMI stores an address and other information associated with the error in MCA::PCS::XGMI::MCA_STATUS_PCS_XGMI. The register is only meaningful if MCA::PCS::XGMI::MCA_STATUS_PCS_XGMI[Val]=1 and MCA::PCS::XGMI::MCA_STATUS_PCS_XGMI[AddrV]=1.

_inst[SERDESBP[3,1]PCSXGMI,SERDESBG[2:0]PCSXGMI,SERDESAG3PCSXGMI]_n[5:0]_aliasMSRLEGACY; MSR0000_047A

_inst[SERDESBP[3,1]PCSXGMI,SERDESBG[2:0]PCSXGMI,SERDESAG3PCSXGMI]_n[5:0]_aliasMSR; MSRC000_21E2

Bits	Description
63:0	ErrorAddr. Read-only. Reset: Cold,0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::PCS::XGMI::MCA_STATUS_PCS_XGMI.

Table 107: MCA_ADDR_PCS_XGMI

Error Type	Bits	Description
DataLossErr	[63:0]	Reserved
TrainingErr	[63:0]	Reserved
FlowCtrlAckErr	[63:0]	Reserved
RxFifoUnderflowErr	[63:0]	Reserved
RxFifoOverflowErr	[63:0]	Reserved
CRCErr	[63:0]	Reserved
BERExceededErr	[63:0]	Reserved
TxMetaDataErr_TxVcidDataErr	[63:0]	Reserved
ReplayBufParityErr	[63:0]	Reserved
DataParityErr	[63:0]	Reserved
ReplayFifoOverflowErr	[63:0]	Reserved
ReplayFifoUnderflowErr	[63:0]	Reserved
ElasticFifoOverflowErr	[63:0]	Reserved
DeskewErr	[63:0]	Reserved
FlowCtrlCRCErr	[63:0]	Reserved
DataStartupLimitErr	[63:0]	Reserved
FCInitTimeoutErr	[63:0]	Reserved
RecoveryTimeoutErr	[63:0]	Reserved
ReadySerialTimeoutErr	[63:0]	Reserved
ReadySerialAttemptErr	[63:0]	Reserved
RecoveryAttemptErr	[63:0]	Reserved
RecoveryRelockAttemptErr	[63:0]	Reserved
ReplayAttemptErr	[63:0]	Reserved
SyncHdrErr	[63:0]	Reserved
TxReplayTimeoutErr	[63:0]	Reserved
RxReplayTimeoutErr	[63:0]	Reserved
LinkSubTxTimeoutErr	[63:0]	Reserved
LinkSubRxTimeoutErr	[63:0]	Reserved
RxCMDPktErr	[63:0]	Reserved
Reserved_29	[63:0]	Reserved
Reserved_30	[63:0]	Reserved

Reserved_31		[63:0]	Reserved
MSR0000_047B...MSRC000_21E3 [PCS_XGMI Machine Check Miscellaneous 0] (MCA::PCS::XGMI::MCA_MISC0_PCS_XGMI)			
Log miscellaneous information associated with errors.			
_inst[SERDESBP[3,1]PCSXGMI,SERDESBG[2:0]PCSXGMI,SERDESAG3PCSXGMI]_n[5:0]_aliasMSRLEGACY; MSR0000_047B			
_inst[SERDESBP[3,1]PCSXGMI,SERDESBG[2:0]PCSXGMI,SERDESAG3PCSXGMI]_n[5:0]_aliasMSR; MSRC000_21E3			
Bits	Description		
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.		
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.		
61	Locked. 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	IntP. 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::PCS::XGMI::MCA_MISC0_PCS_XGMI[Locked]) ? Read-write : Read-only.		
59:56	Reserved.		
55:52	LvtOffset. 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::PCS::XGMI::MCA_MISC0_PCS_XGMI[Locked]) ? Read-write : Read-only.		
51	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn] ! MCA::PCS::XGMI::MCA_MISC0_PCS_XGMI[Locked]) ? Read-write : Read-only.		
50:49	ThresholdIntType. 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::PCS::XGMI::MCA_MISC0_PCS_XGMI[Locked]) ? Read-write : Read-only.		
48	Ovrflw. 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::PCS::XGMI::MCA_MISC0_PCS_XGMI[Locked]) ? Read-write : Read-only.		
47:44	Reserved.		
43:32	ErrCnt. 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::PCS::XGMI::MCA_MISC0_PCS_XGMI[Locked]) ? Read-write : Read-only.		
31:24	BlkPtr. Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.		
23:0	Reserved.		

**MSRC000_21E4 [PCS_XGMI Machine Check Configuration]
(MCA::PCS::XGMI::MCA_CONFIG_PCS_XGMI)**

Reset: 0000_0002_0000_0125h.

Controls configuration of the associated machine check bank.

_inst[SERDESBP[3,1]PCSXGMI,SERDESBG[2:0]PCSXGMI,SERDESAG3PCSXGMI]_n[5:0]_aliasMSR; MSRC000_21E4

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	LogDeferredInMcaStat. Read-write. Reset: 0. Init: BIOS,1. 1=Log deferred errors in MCA::PCS::XGMI::MCA_STATUS_PCS_XGMI and MCA::PCS::XGMI::MCA_ADDR_PCS_XGMI in addition to MCA::PCS::XGMI::MCA_DESTAT_PCS_XGMI and MCA::PCS::XGMI::MCA_DEADDR_PCS_XGMI. 0=Only log deferred errors in MCA::PCS::XGMI::MCA_DESTAT_PCS_XGMI and MCA::PCS::XGMI::MCA_DEADDR_PCS_XGMI. This bit does not affect logging of deferred errors in MCA::PCS::XGMI::MCA_SYND_PCS_XGMI, MCA::PCS::XGMI::MCA_MISC0_PCS_XGMI.
33	Reserved.
32	McaXEnable. 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	IntPresent. Read-only, Volatile . Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::PCS::XGMI::MCA_CONFIG_PCS_XGMI[IntEn].
9	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::PCS::XGMI::MCA_CONFIG_PCS_XGMI[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. Read-only. Reset: 1. 1=MCA::PCS::XGMI::MCA_CONFIG_PCS_XGMI[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::PCS::XGMI::MCA_CONFIG_PCS_XGMI[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	DeferredErrorLoggingSupported. Read-only. Reset: 1. 1=Deferred errors are supported in this MCA bank, and MCA::PCS::XGMI::MCA_CONFIG_PCS_XGMI[LogDeferredInMcaStat] controls the logging behavior of these errors. MCA::PCS::XGMI::MCA_DESTAT_PCS_XGMI and MCA::PCS::XGMI::MCA_DEADDR_PCS_XGMI are supported in this MCA bank. 0=Deferred errors are not supported in this bank.
1	Reserved.
0	McaX. Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::PCS::XGMI::MCA_MISC0_PCS_XGMI[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::PCS::XGMI::MCA_STATUS_PCS_XGMI[TCC] is present.

MSRC000_21E5 [PCS_XGMI IP Identification] (MCA::PCS::XGMI::MCA_IPID_PCS_XGMI)

Reset: 0000_0050_0000_0000h.

The MCA::PCS::XGMI::MCA_IPID_PCS_XGMI register is used by software to determine what IP type and revision is associated with the MCA bank.

_inst[SERDESBP[3,1]PCSXGMI,SERDESBG[2:0]PCSXGMI,SERDESAG3PCSXGMI]_n[5:0]_aliasMSR; MSRC000_21E5

Bits	Description
63:48	McaType. Read-only. Reset: 0000h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi. 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	HardwareID. Read-only. Reset: 050h. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId. Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _instSERDESAG3PCSXGMI_n5_aliasMSR: 1210_9200h
	Init: _instSERDESBG0PCSXGMI_n2_aliasMSR: 11E0_9200h
	Init: _instSERDESBG1PCSXGMI_n3_aliasMSR: 11F0_9200h
	Init: _instSERDESBG2PCSXGMI_n4_aliasMSR: 1200_9200h
	Init: _instSERDESBP1PCSXGMI_n0_aliasMSR: 11B0_9200h
	Init: _instSERDESBP3PCSXGMI_n1_aliasMSR: 11D0_9200h

MSRC000_21E6 [PCS_XGMI Machine Check Syndrome] (MCA::PCS::XGMI::MCA_SYND_PCS_XGMI)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::PCS::XGMI::MCA_STATUS_PCS_XGMI [Thread 0](#)

_inst[SERDESBP[3,1]PCSXGMI,SERDESBG[2:0]PCSXGMI,SERDESAG3PCSXGMI]_n[5:0]_aliasMSR; MSRC000_21E6

Bits	Description
63:32	Syndrom. Read-write, Volatile . Reset: Cold, 0000_0000h. Contains the syndrome, if any, associated with the error logged in MCA::PCS::XGMI::MCA_STATUS_PCS_XGMI. The low-order bit of the syndrome is stored in bit 0, and the syndrome has a length specified by MCA::PCS::XGMI::MCA_SYND_PCS_XGMI[Length]. The Syndrome field is only valid when MCA::PCS::XGMI::MCA_SYND_PCS_XGMI[Length] is not 0.
31:27	Reserved.
26:24	ErrorPriority. Read-write, Volatile . Reset: Cold, 0h. Encodes the priority of the error logged in MCA::PCS::XGMI::MCA_SYND_PCS_XGMI. 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	Length. Read-write, Volatile . Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::PCS::XGMI::MCA_SYND_PCS_XGMI[Syndrome]. Length values greater than 32 (decimal) are interpreted as equal to 32 (decimal). A value of 0 indicates that there is no valid syndrome in MCA::PCS::XGMI::MCA_SYND_PCS_XGMI. For example, a syndrome length of 9 means that MCA::PCS::XGMI::MCA_SYND_PCS_XGMI[Syndrome] bits [8:0] contains a valid syndrome.
17:0	ErrorInformation. Read-write, Volatile . Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 108 [MCA_SYND_PCS_XGMI].

Table 108: MCA_SYND_PCS_XGMI

Error Type	Bits	Description
DataLossErr	[17:0]	Reserved
TrainingErr	[17:0]	Reserved
FlowCtrlAckErr	[17:0]	Reserved
RxFifoUnderflowErr	[17:0]	Reserved
RxFifoOverflowErr	[17:0]	Reserved
CRCErr	[17:0]	Reserved
BERExceededErr	[17:0]	Reserved
TxMetaDataErr_TxVcidDataErr	[17:0]	Reserved
ReplayBufParityErr	[17:0]	Reserved

DataParityErr	[17:0]	Reserved
ReplayFifoOverflowErr	[17:0]	Reserved
ReplayFifoUnderflowErr	[17:0]	Reserved
ElasticFifoOverflowErr	[17:0]	Reserved
DeskewErr	[17:0]	Reserved
FlowCtrlCRCErr	[17:0]	Reserved
DataStartupLimitErr	[17:0]	Reserved
FCInitTimeoutErr	[17:0]	Reserved
RecoveryTimeoutErr	[17:0]	Reserved
ReadySerialTimeoutErr	[17:0]	Reserved
ReadySerialAttemptErr	[17:0]	Reserved
RecoveryAttemptErr	[17:0]	Reserved
RecoveryRelockAttemptErr	[17:0]	Reserved
ReplayAttemptErr	[17:0]	Reserved
SyncHdrErr	[17:0]	Reserved
TxReplayTimeoutErr	[17:0]	Reserved
RxReplayTimeoutErr	[17:0]	Reserved
LinkSubTxTimeoutErr	[17:0]	Reserved
LinkSubRxTimeoutErr	[17:0]	Reserved
RxCMDPktErr	[17:0]	Reserved
Reserved_29	[17:0]	Reserved
Reserved_30	[17:0]	Reserved
Reserved_31	[17:0]	Reserved

MSRC000_21E8 [PCS_XGMI Machine Check Deferred Error Status] (MCA::PCS::XGMI::MCA_DESTAT_PCS_XGMI)

Reset: Cold,0000_0000_0000_0000h.

Holds status information for the first deferred error seen in this bank.

_inst[SERDESBP[3,1]PCSXGMI,SERDESBG[2:0]PCSXGMI,SERDESAG3PCSXGMI]_n[5:0]_aliasMSR; MSRC000_21E8

Bits	Description
63	Val. Read-write, Volatile . Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	Overflow. 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 overwrite priorities.)
61:59	RESERV4. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	AddrV. Read-write, Volatile . Reset: Cold,0. 1=MCA::PCS::XGMI::MCA_DEADDR_PCS_XGMI contains address information associated with the error.
57:54	RESERV3. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	SyndV. Read-write, Volatile . Reset: Cold,0. 1=This error logged information in MCA::PCS::XGMI::MCA_SYND_PCS_XGMI. If MCA::PCS::XGMI::MCA_SYND_PCS_XGMI[ErrorPriority] is the same as the priority of the error in MCA::PCS::XGMI::MCA_STATUS_PCS_XGMI, then the information in MCA::PCS::XGMI::MCA_SYND_PCS_XGMI is associated with the error in MCA::PCS::XGMI::MCA_DESTAT_PCS_XGMI.
52:45	RESERV2. Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	Deferred. 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	RESERV1. Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	AddrLsb. Read-write, Volatile . Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::PCS::XGMI::MCA_ADDR_PCS_XGMI[ErrorAddr]. A value of 0 indicates that MCA::PCS::XGMI::MCA_ADDR_PCS_XGMI[63:0] contains a valid byte address. A value of 6 indicates that MCA::PCS::XGMI::MCA_ADDR_PCS_XGMI[63:6] contains a valid cache line address and that MCA::PCS::XGMI::MCA_ADDR_PCS_XGMI[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::PCS::XGMI::MCA_ADDR_PCS_XGMI[63:12] contain a valid 4KB memory page and that MCA::PCS::XGMI::MCA_ADDR_PCS_XGMI[11:0] should be ignored by error handling software.
23:22	RESERV0. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	ErrorCodeExt. 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	ErrorCode. Read-write, Volatile . Reset: Cold,0000h. Error code for this error.

MSRC000_21E9 [PCS_XGMI Deferred Error Address] (MCA::PCS::XGMI::MCA_DEADDR_PCS_XGMI)

Read-only. Reset: Cold,0000_0000_0000_0000h.

The MCA::PCS::XGMI::MCA_DEADDR_PCS_XGMI register stores the address associated with the error in MCA::PCS::XGMI::MCA_DESTAT_PCS_XGMI. The register is only meaningful if MCA::PCS::XGMI::MCA_DESTAT_PCS_XGMI[Val]=1 and MCA::PCS::XGMI::MCA_DESTAT_PCS_XGMI[AddrV]=1. The lowest valid bit of the address is defined by MCA::PCS::XGMI::MCA_DESTAT_PCS_XGMI[AddrLsb].

_inst[SERDESBP[3,1]PCSXGMI,SERDESBG[2:0]PCSXGMI,SERDESAG3PCSXGMI]_n[5:0]_aliasMSR; MSRC000_21E9

Bits	Description
63:0	ErrorAddr. Read-only. Reset: Cold,0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::PCS::XGMI::MCA_DESTAT_PCS_XGMI.

**MSRC001_041E [PCS_XGMI Machine Check Control Mask]
(MCA::PCS::XGMI::MCA_CTL_MASK_PCS_XGMI)**

Read-write. Reset: 0000_0000_0000_0000h.

Inhibit detection of an error source.

_inst[SERDESBP[3,1]PCSXGMI,SERDESBG[2:0]PCSXGMI,SERDESAG3PCSXGMI]_n[5:0]_aliasMSR; MSRC001_041E

Bits	Description
63:29	Reserved.
28	RxCMDPktErr. Read-write. Reset: 0. RxCMDPktErr for xGMI3
27	LinkSubRxTimeoutErr. Read-write. Reset: 0. LinkSubRxTimeoutErr for xGMI3
26	LinkSubTxTimeoutErr. Read-write. Reset: 0. LinkSubTxTimeoutErr for xGMI3
25	RxReplayTimeoutErr. Read-write. Reset: 0. RxReplayTimeoutErr for xGMI3
24	TxReplayTimeoutErr. Read-write. Reset: 0. TxReplayTimeoutErr for xGMI3
23	SyncHdrErr. Read-write. Reset: 0. SyncHdrErr for xGMI3
22	ReplayAttemptErr. Read-write. Reset: 0. ReplayAttemptErr for xGMI3
21	RecoveryRelockAttemptErr. Read-write. Reset: 0. RecoveryRelockAttemptErr
20	RecoveryAttemptErr. Read-write. Reset: 0. RecoveryAttemptErr
19	ReadySerialAttemptErr. Read-write. Reset: 0. ReadySerialAttemptErr
18	ReadySerialTimeoutErr. Read-write. Reset: 0. ReadySerialTimeoutErr
17	RecoveryTimeoutErr. Read-write. Reset: 0. RecoveryTimeoutErr
16	FCInitTimeoutErr. Read-write. Reset: 0. FCInitTimeoutErr
15	DataStartupLimitErr. Read-write. Reset: 0. DataStartupLimitErr
14	FlowCtrlCRCErr. Read-write. Reset: 0. FlowCtrlCRCErr for xGMI3
13	DeskewErr. Read-write. Reset: 0. DeskewErr
12	ElasticFifoOverflowErr. Read-write. Reset: 0. ElasticFifoOverflowErr
11	ReplayFifoUnderflowErr. Read-write. Reset: 0. ReplayFifoUnderflowErr
10	ReplayFifoOverflowErr. Read-write. Reset: 0. ReplayFifoOverflowErr
9	DataParityErr. Read-write. Reset: 0. DataParityErr
8	ReplayBufParityErr. Read-write. Reset: 0. ReplayBufParityErr
7	TxMetaDataErr_TxVcidDataErr. Read-write. Reset: 0. TxVcidDataErr for xGMI3.
6	BERExceededErr. Read-write. Reset: 0. BERExceededErr
5	CRCErr. Read-write. Reset: 0. CRCErr
4	RxFifoOverflowErr. Read-write. Reset: 0. RxFifoOverflowErr for xGMI3
3	RxFifoUnderflowErr. Read-write. Reset: 0. RxFifoUnderflowErr for xGMI3
2	FlowCtrlAckErr. Read-write. Reset: 0. FlowCtrlAckErr for xGMI3.
1	TrainingErr. Read-write. Reset: 0. TrainingErr.
0	DataLossErr. Read-write. Reset: 0. DataLossErr

**MSRC000_21EE [PCS_XGMI Machine Check Syndrome Extended]
(MCA::PCS::XGMI::MCA_SYND1_PCS_XGMI)**

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::PCS::XGMI::MCA_STATUS_PCS_XGMI [Thread 0](#)

_inst[SERDESBP[3,1]PCSXGMI,SERDESBG[2:0]PCSXGMI,SERDESAG3PCSXGMI]_n[5:0]_aliasMSR; MSRC000_21EE

Bits	Description
63:0	Syndrome. Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::PCS::XGMI::MCA_SYND1_PCS_XGMI register stores information associated with the error in MCA::PCS::XGMI::MCA_STATUS_PCS_XGMI or MCA_DESTAT. The register is meaningful if MCA::PCS::XGMI::MCA_STATUS_PCS_XGMI[SyndV]=1. When MCA::PCS::XGMI::MCA_CONFIG_PCS_XGMI[McaFruTextInMca]=1, MCA::PCS::XGMI::MCA_SYND1_PCS_XGMI stores ASCII FruText associated with the error.

**MSRC000_21EF [PCS_XGMI Machine Check Syndrome Extended]
(MCA::PCS::XGMI::MCA_SYND2_PCS_XGMI)**

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::PCS::XGMI::MCA_STATUS_PCS_XGMI [Thread 0](#)
_inst[SERDESBP[3,1]PCSXGMI,SERDESBG[2:0]PCSXGMI,SERDESAG3PCSXGMI]_n[5:0]_aliasMSR; MSRC000_21EF

Bits	Description
63:0	Syndrome. Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::PCS::XGMI::MCA_SYND2_PCS_XGMI register stores information associated with the error in MCA::PCS::XGMI::MCA_STATUS_PCS_XGMI or MCA_DESTAT. The register is meaningful if MCA::PCS::XGMI::MCA_STATUS_PCS_XGMI[SyndV]=1. When MCA::PCS::XGMI::MCA_CONFIG_PCS_XGMI[McaFruTextInMca]=1, MCA::PCS::XGMI::MCA_SYND2_PCS_XGMI stores ASCII FruText associated with the error.

3.2.5.20 NBIF
MSR0000_0470...MSRC000_21C0 [NBIF Machine Check Control] (MCA::NBIF::MCA_CTL_NBIF)

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::NBIF::MCA_CTL_NBIF register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG_CTL. Does not affect error detection, correction, or logging.

_ch0_inst[NBIF[2:0]]_n[5:0]_nbio[1:0]_aliasMSRLEGACY; MSR0000_0470

_ch0_inst[NBIF[2:0]]_n[5:0]_nbio[1:0]_aliasMSR; MSRC000_21C0

Bits	Description
63:4	Reserved.
3	SDP_PARITY_ERR. Read-write. Reset: 0. SDP Parity error was detected!
2	NTB_ERR_EVENT. Read-write. Reset: 0. NTB ERROR EVENT was detected!
1	SRAM_ECC_ERR. Read-write. Reset: 0. An SRAM ECC error was detected!
0	TIMEOUT_ERR. Read-write. Reset: 0. A Timeout error from gmi was detected!

MSR0000_0471...MSRC000_21C1 [NBIF Machine Check Status] (MCA::NBIF::MCA_STATUS_NBIF)

Reset: Cold,0000_0000_0000_0000h.

Logs information associated with errors.

_ch0_inst[NBIF[2:0]]_n[5:0]_nbio[1:0]_aliasMSRLEGACY; MSR0000_0471

_ch0_inst[NBIF[2:0]]_n[5:0]_nbio[1:0]_aliasMSR; MSRC000_21C1

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::NBIF::MCA_CTL_NBIF. This bit is a copy of bit in MCA::NBIF::MCA_CTL_NBIF for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	MiscV. Reset: Cold,0. 1=Valid thresholding in MCA::NBIF::MCA_MISC0_NBIF. 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	AddrV. Reset: Cold,0. 1=MCA::NBIF::MCA_ADDR_NBIF contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	PCC. 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	ErrCoreIdVal. 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	TCC. 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::NBIF::MCA_STATUS_NBIF[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	RESERV54. 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	SyndV. Reset: Cold,0. 1=This error logged information in MCA::NBIF::MCA_SYND_NBIF. If MCA::NBIF::MCA_SYND_NBIF[ErrorPriority] is the same as the priority of the error in MCA::NBIF::MCA_STATUS_NBIF, then the information in MCA::NBIF::MCA_SYND_NBIF is associated with the error in MCA::NBIF::MCA_STATUS_NBIF. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::NBIF::MCA_ADDR_NBIF[ErrorAddr]. A value of 0 indicates that MCA::NBIF::MCA_ADDR_NBIF[63:0] contains a valid byte address. A value of 6 indicates that MCA::NBIF::MCA_ADDR_NBIF[63:6] contains a valid cache line address and that MCA::NBIF::MCA_ADDR_NBIF[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::NBIF::MCA_ADDR_NBIF[63:12] contain a valid 4KB memory page and that MCA::NBIF::MCA_ADDR_NBIF[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	RESERV22. 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	ErrorCodeExt. 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::NBIF::MCA_CTL_NBIF 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	ErrorCode. 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 109: MCA_STATUS_NBIF

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
TIMEOUT_ERR	0x0	1	1	1	0	0	0
SRAM_ECC_ERR	0x1	0/1	0/1	0/1	0	0	0
NTB_ERR_EVENT	0x2	1	1	1	0	0	0
SDP_PARITY_ERR	0x3	0	0	0	1	0	0

MSR0000_0472...MSRC000_21C2 [NBIF Machine Check Address] (MCA::NBIF::MCA_ADDR_NBIF)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

MCA::NBIF::MCA_ADDR_NBIF stores an address and other information associated with the error in MCA::NBIF::MCA_STATUS_NBIF. The register is only meaningful if MCA::NBIF::MCA_STATUS_NBIF[Val]=1 and MCA::NBIF::MCA_STATUS_NBIF[AddrV]=1.

_ch0_inst[NBIF[2:0]]_n[5:0]_nbio[1:0]_aliasMSRLEGACY; MSR0000_0472

_ch0_inst[NBIF[2:0]]_n[5:0]_nbio[1:0]_aliasMSR; MSRC000_21C2

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::NBIF::MCA_STATUS_NBIF. For physical addresses, the most significant bit is given by Core::X86::Cpuid::LongModeInfo[PhysAddrSize].

Table 110: MCA_ADDR_NBIF

Error Type	Bits	Description
TIMEOUT_ERR	[63:0]	Reserved
SRAM_ECC_ERR	[63:0]	Reserved
NTB_ERR_EVENT	[63:0]	Reserved
SDP_PARITY_ERR	[63:0]	Reserved

MSR0000_0473...MSRC000_21C3 [NBIF Machine Check Miscellaneous 0] (MCA::NBIF::MCA_MISC0_NBIF)

Log miscellaneous information associated with errors.

_ch0_inst[NBIF[2:0]]_n[5:0]_nbio[1:0]_aliasMSRLEGACY; MSR0000_0473

_ch0_inst[NBIF[2:0]]_n[5:0]_nbio[1:0]_aliasMSR; MSRC000_21C3

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr:: HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr:: HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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::NBIF::MCA_MISC0_NBIF[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	LvtOffset. 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::NBIF::MCA_MISC0_NBIF[Locked]) ? Read-write : Read-only.
51	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr:: HWCR[McStatusWrEn] !MCA::NBIF::MCA_MISC0_NBIF[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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::NBIF::MCA_MISC0_NBIF[Locked]) ? Read-write : Read-only.
48	Ovrflw. 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::NBIF::MCA_MISC0_NBIF[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	ErrCnt. 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::NBIF::MCA_MISC0_NBIF[Locked]) ? Read-write : Read-only.
31:24	BlkPtr. Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

MSRC000_21C4 [NBIF Machine Check Configuration] (MCA::NBIF::MCA_CONFIG_NBIF)

Reset: 0000_0002_0000_0125h.

Controls configuration of the associated machine check bank.

_ch0_inst[NBIF[2:0]]_n[5:0]_nbio[1:0]_aliasMSR; MSRC000_21C4

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	LogDeferredInMcaStat. Read-write. Reset: 0. Init: BIOS,1. 1=Log deferred errors in MCA::NBIF::MCA_STATUS_NBIF and MCA::NBIF::MCA_ADDR_NBIF in addition to MCA::NBIF::MCA_DESTAT_NBIF and MCA::NBIF::MCA_DEADDR_NBIF. 0=Only log deferred errors in MCA::NBIF::MCA_DESTAT_NBIF and MCA::NBIF::MCA_DEADDR_NBIF. This bit does not affect logging of deferred errors in MCA::NBIF::MCA_SYND_NBIF, MCA::NBIF::MCA_MISC0_NBIF.
33	Reserved.
32	McaXEnable. 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	IntPresent. Read-only, Volatile . Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::NBIF::MCA_CONFIG_NBIF[IntEn].
9	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::NBIF::MCA_CONFIG_NBIF[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. Read-only. Reset: 1. 1=MCA::NBIF::MCA_CONFIG_NBIF[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::NBIF::MCA_CONFIG_NBIF[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	DeferredErrorLoggingSupported. Read-only. Reset: 1. 1=Deferred errors are supported in this MCA bank, and MCA::NBIF::MCA_CONFIG_NBIF[LogDeferredInMcaStat] controls the logging behavior of these errors. MCA::NBIF::MCA_DESTAT_NBIF and MCA::NBIF::MCA_DEADDR_NBIF are supported in this MCA bank. 0=Deferred errors are not supported in this bank.
1	Reserved.
0	McaX. Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::NBIF::MCA_MISC0_NBIF[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::NBIF::MCA_STATUS_NBIF[TCC] is present.

MSRC000_21C5 [NBIF IP Identification] (MCA::NBIF::MCA_IPID_NBIF)

Reset: 0000_006C_0000_0000h.

The MCA::NBIF::MCA_IPID_NBIF register is used by software to determine what IP type and revision is associated with the MCA bank.

_ch0_inst[NBIF[2:0]]_n[5:0]_nbio[1:0]_aliasMSR; MSRC000_21C5

Bits	Description
63:48	McaType. Read-only. Reset: 0000h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi. 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	HardwareID. Read-only. Reset: 06Ch. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId. Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _ch0_instNBIF0_n0_nbio0_aliasMSR: 1013_C000h
	Init: _ch0_instNBIF0_n3_nbio1_aliasMSR: 1033_C000h
	Init: _ch0_instNBIF1_n1_nbio0_aliasMSR: 1023_C000h
	Init: _ch0_instNBIF1_n4_nbio1_aliasMSR: 1043_C000h
	Init: _ch0_instNBIF2_n2_nbio0_aliasMSR: 1053_C000h
	Init: _ch0_instNBIF2_n5_nbio1_aliasMSR: 1073_C000h

MSRC000_21C6 [NBIF Machine Check Syndrome] (MCA::NBIF::MCA_SYND_NBIF)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::NBIF::MCA_STATUS_NBIF [Thread 0](#)

_ch0_inst[NBIF[2:0]]_n[5:0]_nbio[1:0]_aliasMSR; MSRC000_21C6

Bits	Description
63:32	Syndrome. Read-write, Volatile . Reset: Cold, 0000_0000h. Contains the syndrome, if any, associated with the error logged in MCA::NBIF::MCA_STATUS_NBIF. The low-order bit of the syndrome is stored in bit 0, and the syndrome has a length specified by MCA::NBIF::MCA_SYND_NBIF[Length]. The Syndrome field is only valid when MCA::NBIF::MCA_SYND_NBIF[Length] is not 0.
31:27	Reserved.
26:24	ErrorPriority. Read-write, Volatile . Reset: Cold, 0h. Encodes the priority of the error logged in MCA::NBIF::MCA_SYND_NBIF. 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	Length. Read-write, Volatile . Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::NBIF::MCA_SYND_NBIF[Syndrome]. Length values greater than 32 (decimal) are interpreted as equal to 32 (decimal). A value of 0 indicates that there is no valid syndrome in MCA::NBIF::MCA_SYND_NBIF. For example, a syndrome length of 9 means that MCA::NBIF::MCA_SYND_NBIF[Syndrome] bits [8:0] contains a valid syndrome.
17:0	ErrorInformation. Read-write, Volatile . Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 111 [MCA_SYND_NBIF].

Table 111: MCA_SYND_NBIF

Error Type	Bits	Description
TIMEOUT_ERR	[10:0] [13:11] [16:14] [17]	SDP transaction Req_unitID leaf ID of detected RAS leaf RAS central ID this error whether is from response transaction
SRAM_ECC_ERR	[10:0] [13:11] [16:14] [17]	SDP transaction Req_unitID leaf ID of detected RAS leaf RAS central ID this error whether is from response transaction
NTB_ERR_EVENT	[10:0] [14:11]	SDP transaction Req_unitID SDP transaction ReqUser bits

SDP_PARITY_ERR	[10:0] [13:11] [16:14] [17]	SDP transaction Req_unitID leaf ID of detected RAS leaf RAS central ID this error whether is from response transaction
----------------	--------------------------------------	---

MSRC000_21C8 [NBIF Machine Check Deferred Error Status] (MCA::NBIF::MCA_DESTAT_NBIF)

Reset: Cold,0000_0000_0000_0000h.

Holds status information for the first deferred error seen in this bank.

_ch0_inst[NBIF[2:0]]_n[5:0]_nbio[1:0]_aliasMSR; MSRC000_21C8

Bits	Description
63	Val. Read-write, Volatile . Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	Overflow. 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 overwrite priorities.)
61:59	RESERV4. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	AddrV. Read-write, Volatile . Reset: Cold,0. 1=MCA::NBIF::MCA_DEADDR_NBIF contains address information associated with the error.
57:54	RESERV3. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	SyndV. Read-write, Volatile . Reset: Cold,0. 1=This error logged information in MCA::NBIF::MCA_SYND_NBIF. If MCA::NBIF::MCA_SYND_NBIF[ErrorPriority] is the same as the priority of the error in MCA::NBIF::MCA_STATUS_NBIF, then the information in MCA::NBIF::MCA_SYND_NBIF is associated with the error in MCA::NBIF::MCA_DESTAT_NBIF.
52:45	RESERV2. Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	Deferred. 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	RESERV1. Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	AddrLsb. Read-write, Volatile . Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::NBIF::MCA_ADDR_NBIF[ErrorAddr]. A value of 0 indicates that MCA::NBIF::MCA_ADDR_NBIF[63:0] contains a valid byte address. A value of 6 indicates that MCA::NBIF::MCA_ADDR_NBIF[63:6] contains a valid cache line address and that MCA::NBIF::MCA_ADDR_NBIF[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::NBIF::MCA_ADDR_NBIF[63:12] contain a valid 4KB memory page and that MCA::NBIF::MCA_ADDR_NBIF[11:0] should be ignored by error handling software.
23:22	RESERV0. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	ErrorCodeExt. 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	ErrorCode. Read-write, Volatile . Reset: Cold,0000h. Error code for this error.

MSRC000_21C9 [NBIF Deferred Error Address] (MCA::NBIF::MCA_DEADDR_NBIF)

Read-write, [Volatile](#). Reset: Cold,0000_0000_0000_0000h.

The MCA::NBIF::MCA_DEADDR_NBIF register stores the address associated with the error in MCA::NBIF::MCA_DESTAT_NBIF. The register is only meaningful if MCA::NBIF::MCA_DESTAT_NBIF[Val]=1 and MCA::NBIF::MCA_DESTAT_NBIF[AddrV]=1. The lowest valid bit of the address is defined by MCA::NBIF::MCA_DESTAT_NBIF[AddrLsb].

_ch0_inst[NBIF[2:0]]_n[5:0]_nbio[1:0]_aliasMSR; MSRC000_21C9

Bits	Description
63:0	ErrorAddr. Read-write, Volatile . Reset: Cold,0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::NBIF::MCA_DESTAT_NBIF. The lowest-order valid bit of the address is specified in MCA::NBIF::MCA_DESTAT_NBIF[AddrLsb].

MSRC001_041C [NBIF Machine Check Control Mask] (MCA::NBIF::MCA_CTL_MASK_NBIF)

Read-write. Reset: 0000_0000_0000_0000h.

Inhibit detection of an error source.

_ch0_inst[NBIF[2:0]]_n[5:0]_nbio[1:0]_aliasMSR; MSRC001_041C

Bits	Description
63:4	Reserved.
3	SDP_PARITY_ERR. Read-write. Reset: 0. SDP Parity error was detected!
2	NTB_ERR_EVENT. Read-write. Reset: 0. NTB ERROR EVENT was detected!
1	SRAM_ECC_ERR. Read-write. Reset: 0. An SRAM ECC error was detected!
0	TIMEOUT_ERR. Read-write. Reset: 0. A Timeout error from gmi was detected!

MSRC000_21CE [NBIF Machine Check Syndrome Extended] (MCA::NBIF::MCA_SYND1_NBIF)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::NBIF::MCA_STATUS_NBIF [Thread](#) 0

_ch0_inst[NBIF[2:0]]_n[5:0]_nbio[1:0]_aliasMSR; MSRC000_21CE

Bits	Description
63:0	Syndrome. Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::NBIF::MCA_SYND1_NBIF register stores information associated with the error in MCA::NBIF::MCA_STATUS_NBIF or MCA_DESTAT. The register is meaningful if MCA::NBIF::MCA_STATUS_NBIF[SyndV]=1. When MCA::NBIF::MCA_CONFIG_NBIF[McaFruTextInMca]=1, MCA::NBIF::MCA_SYND1_NBIF stores ASCII FruText associated with the error.

MSRC000_21CF [NBIF Machine Check Syndrome Extended] (MCA::NBIF::MCA_SYND2_NBIF)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::NBIF::MCA_STATUS_NBIF [Thread](#) 0

_ch0_inst[NBIF[2:0]]_n[5:0]_nbio[1:0]_aliasMSR; MSRC000_21CF

Bits	Description
63:0	Syndrome. Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::NBIF::MCA_SYND2_NBIF register stores information associated with the error in MCA::NBIF::MCA_STATUS_NBIF or MCA_DESTAT. The register is meaningful if MCA::NBIF::MCA_STATUS_NBIF[SyndV]=1. When MCA::NBIF::MCA_CONFIG_NBIF[McaFruTextInMca]=1, MCA::NBIF::MCA_SYND2_NBIF stores ASCII FruText associated with the error.

3.2.5.21 SHUB**MSR0000_0474...MSRC000_21D0 [SHUB Machine Check Control] (MCA::SHUB::MCA_CTL_SHUB)**

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::SHUB::MCA_CTL_SHUB register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG_CTL. Does not affect error detection, correction, or logging.

_ch1_inst[NBIF[1:0]SHUB0]_n[3:0]_nbio[1:0]_aliasMSRLEGACY; MSR0000_0474

_ch1_inst[NBIF[1:0]SHUB0]_n[3:0]_nbio[1:0]_aliasMSR; MSRC000_21D0

Bits	Description
63:4	Reserved.
3	SDP_PARITY_ERR. Read-write. Reset: 0. Reset 0. SDP Parity error was detected!
2	NTB_ERR_EVENT. Read-write. Reset: 0. Reset 0. NTB ERROR EVENT was detected!
1	SRAM_ECC_ERR. Read-write. Reset: 0. Reset 0. An SRAM ECC error was detected!
0	NA. Read-write. Reset: 0. Reset 0. no use

MSR0000_0475...MSRC000_21D1 [SHUB Machine Check Status] (MCA::SHUB::MCA_STATUS_SHUB)

Reset: Cold,0000_0000_0000_0000h.

Logs information associated with errors.

_ch1_inst[NBIF[1:0]SHUB0]_n[3:0]_nbio[1:0]_aliasMSRLEGACY; MSR0000_0475

_ch1_inst[NBIF[1:0]SHUB0]_n[3:0]_nbio[1:0]_aliasMSR; MSRC000_21D1

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::SHUB::MCA_CTL_SHUB. This bit is a copy of bit in MCA::SHUB::MCA_CTL_SHUB for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	MiscV. Reset: Cold,0. 1=Valid thresholding in MCA::SHUB::MCA_MISC0_SHUB. 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	AddrV. Reset: Cold,0. 1=MCA::SHUB::MCA_ADDR_SHUB contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	PCC. 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	ErrCoreIdVal. 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	TCC. 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::SHUB::MCA_STATUS_SHUB[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	RESERV54. 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	SyndV. Reset: Cold,0. 1=This error logged information in MCA::SHUB::MCA_SYND_SHUB. If MCA::SHUB::MCA_SYND_SHUB[ErrorPriority] is the same as the priority of the error in MCA::SHUB::MCA_STATUS_SHUB, then the information in MCA::SHUB::MCA_SYND_SHUB is associated with the error in MCA::SHUB::MCA_STATUS_SHUB. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::SHUB::MCA_ADDR_SHUB[ErrorAddr]. A value of 0 indicates that MCA::SHUB::MCA_ADDR_SHUB[63:0] contains a valid byte address. A value of 6 indicates that MCA::SHUB::MCA_ADDR_SHUB[63:6] contains a valid cache line address and that MCA::SHUB::MCA_ADDR_SHUB[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::SHUB::MCA_ADDR_SHUB[63:12] contain a valid 4KB memory page and that MCA::SHUB::MCA_ADDR_SHUB[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	RESERV22. 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	ErrorCodeExt. 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::SHUB::MCA_CTL_SHUB 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	ErrorCode. 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 112: MCA_STATUS_SHUB

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
Reserved	0x0					0	0
SRAM_ECC_ERR	0x1	0/1	0/1	0/1	0	0	0
NTB_ERR_EVENT	0x2	1	1	1	0	0	0
SDP_PARITY_ERR	0x3	0	0	0	1	0	0

MSR0000_0476...MSRC000_21D2 [SHUB Machine Check Address] (MCA::SHUB::MCA_ADDR_SHUB)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

MCA::SHUB::MCA_ADDR_SHUB stores an address and other information associated with the error in MCA::SHUB::MCA_STATUS_SHUB. The register is only meaningful if MCA::SHUB::MCA_STATUS_SHUB[Val]=1 and MCA::SHUB::MCA_STATUS_SHUB[AddrV]=1.

_ch1_inst[NBIF[1:0]SHUB0]_n[3:0]_nbio[1:0]_aliasMSRLEGACY; MSR0000_0476

_ch1_inst[NBIF[1:0]SHUB0]_n[3:0]_nbio[1:0]_aliasMSR; MSRC000_21D2

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::SHUB::MCA_STATUS_SHUB. For physical addresses, the most significant bit is given by Core::X86::Cpuid::LongModeInfo[PhysAddrSize].

Table 113: MCA_ADDR_SHUB

Error Type	Bits	Description
Reserved	[63:0]	Reserved
SRAM_ECC_ERR	[63:0]	Reserved
NTB_ERR_EVENT	[63:0]	Reserved
SDP_PARITY_ERR	[63:0]	Reserved

**MSR0000_0477...MSRC000_21D3 [SHUB Machine Check Miscellaneous 0]
(MCA::SHUB::MCA_MISC0_SHUB)**

Log miscellaneous information associated with errors.

_ch1_inst[NBIF[1:0]SHUB0]_n[3:0]_nbio[1:0]_aliasMSRLEGACY; MSR0000_0477

_ch1_inst[NBIF[1:0]SHUB0]_n[3:0]_nbio[1:0]_aliasMSR; MSRC000_21D3

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msrr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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::SHUB::MCA_MISC0_SHUB[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	LvtOffset. 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::SHUB::MCA_MISC0_SHUB[Locked]) ? Read-write : Read-only.
51	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msrr::HWCR[McStatusWrEn] !MCA::SHUB::MCA_MISC0_SHUB[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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::SHUB::MCA_MISC0_SHUB[Locked]) ? Read-write : Read-only.
48	Ovrflw. 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::SHUB::MCA_MISC0_SHUB[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	ErrCnt. 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::SHUB::MCA_MISC0_SHUB[Locked]) ? Read-write : Read-only.
31:24	BlkPtr. Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

MSRC000_21D4 [SHUB Machine Check Configuration] (MCA::SHUB::MCA_CONFIG_SHUB)

Reset: 0000_0002_0000_0125h.

Controls configuration of the associated machine check bank.

_ch1_inst[NBIF[1:0]SHUB0]_n[3:0]_nbio[1:0]_aliasMSR; MSRC000_21D4

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	LogDeferredInMcaStat. Read-write. Reset: 0. Init: BIOS,1. 1=Log deferred errors in MCA::SHUB::MCA_STATUS_SHUB and MCA::SHUB::MCA_ADDR_SHUB in addition to MCA::SHUB::MCA_DESTAT_SHUB and MCA::SHUB::MCA_DEADDR_SHUB. 0=Only log deferred errors in MCA::SHUB::MCA_DESTAT_SHUB and MCA::SHUB::MCA_DEADDR_SHUB. This bit does not affect logging of deferred errors in MCA::SHUB::MCA_SYND_SHUB, MCA::SHUB::MCA_MISC0_SHUB.
33	Reserved.
32	McaXEnable. 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	IntPresent. Read-only, Volatile . Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::SHUB::MCA_CONFIG_SHUB[IntEn].
9	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::SHUB::MCA_CONFIG_SHUB[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. Read-only. Reset: 1. 1=MCA::SHUB::MCA_CONFIG_SHUB[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::SHUB::MCA_CONFIG_SHUB[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	DeferredErrorLoggingSupported. Read-only. Reset: 1. 1=Deferred errors are supported in this MCA bank, and MCA::SHUB::MCA_CONFIG_SHUB[LogDeferredInMcaStat] controls the logging behavior of these errors. MCA::SHUB::MCA_DESTAT_SHUB and MCA::SHUB::MCA_DEADDR_SHUB are supported in this MCA bank. 0=Deferred errors are not supported in this bank.
1	Reserved.
0	McaX. Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::SHUB::MCA_MISC0_SHUB[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::SHUB::MCA_STATUS_SHUB[TCC] is present.

MSRC000_21D5 [SHUB IP Identification] (MCA::SHUB::MCA_IPID_SHUB)

Reset: 0000_0080_0000_0000h.

The MCA::SHUB::MCA_IPID_SHUB register is used by software to determine what IP type and revision is associated with the MCA bank.

_ch1_inst[NBIF[1:0]SHUB0]_n[3:0]_nbio[1:0]_aliasMSR; MSRC000_21D5

Bits	Description
63:48	McaType . Read-only. Reset: 0000h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi . 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	HardwareID . Read-only. Reset: 080h. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _ch1_instNBIF0SHUB0_n0_nbio0_aliasMSR: 0141_C000h
	Init: _ch1_instNBIF0SHUB0_n2_nbio1_aliasMSR: 0161_C000h
	Init: _ch1_instNBIF1SHUB0_n1_nbio0_aliasMSR: 0151_C000h
	Init: _ch1_instNBIF1SHUB0_n3_nbio1_aliasMSR: 0171_C000h

MSRC000_21D6 [SHUB Machine Check Syndrome] (MCA::SHUB::MCA_SYND_SHUB)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::SHUB::MCA_STATUS_SHUB [Thread 0](#)

_ch1_inst[NBIF[1:0]SHUB0]_n[3:0]_nbio[1:0]_aliasMSR; MSRC000_21D6

Bits	Description
63:32	Syndrom . Read-write, Volatile . Reset: Cold, 0000_0000h. Contains the syndrome, if any, associated with the error logged in MCA::SHUB::MCA_STATUS_SHUB. The low-order bit of the syndrome is stored in bit 0, and the syndrome has a length specified by MCA::SHUB::MCA_SYND_SHUB[Length]. The Syndrome field is only valid when MCA::SHUB::MCA_SYND_SHUB[Length] is not 0.
31:27	Reserved.
26:24	ErrorPriority . Read-write, Volatile . Reset: Cold, 0h. Encodes the priority of the error logged in MCA::SHUB::MCA_SYND_SHUB. 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	Length . Read-write, Volatile . Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::SHUB::MCA_SYND_SHUB[Syndrome]. Length values greater than 32 (decimal) are interpreted as equal to 32 (decimal). A value of 0 indicates that there is no valid syndrome in MCA::SHUB::MCA_SYND_SHUB. For example, a syndrome length of 9 means that MCA::SHUB::MCA_SYND_SHUB[Syndrome] bits [8:0] contains a valid syndrome.
17:0	ErrorInformation . Read-write, Volatile . Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 114 [MCA_SYND_SHUB].

Table 114: MCA_SYND_SHUB

Error Type	Bits	Description
Reserved	[17:0]	Reserved
SRAM_ECC_ERR	[10:0] [13:11] [16:14] [17]	SDP transaction Req_unitID leaf ID of detected RAS leaf RAS central ID this error whether is from response transaction
NTB_ERR_EVENT	[10:0] [14:11]	SDP transaction Req_unitID SDP transaction ReqUser bits
SDP_PARITY_ERR	[10:0] [13:11] [16:14] [17]	SDP transaction Req_unitID leaf ID of detected RAS leaf RAS central ID this error whether is from response transaction

MSRC000_21D8 [SHUB Machine Check Deferred Error Status] (MCA::SHUB::MCA_DESTAT_SHUB)

Reset: Cold,0000_0000_0000_0000h.

Holds status information for the first deferred error seen in this bank.

_ch1_inst[NBIF[1:0]SHUB0]_n[3:0]_nbio[1:0]_aliasMSR; MSRC000_21D8

Bits	Description
63	Val. Read-write, Volatile . Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	Overflow. 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 overwrite priorities.)
61:59	RESERV4. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	AddrV. Read-write, Volatile . Reset: Cold,0. 1=MCA::SHUB::MCA_DEADDR_SHUB contains address information associated with the error.
57:54	RESERV3. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	SyndV. Read-write, Volatile . Reset: Cold,0. 1=This error logged information in MCA::SHUB::MCA_SYND_SHUB. If MCA::SHUB::MCA_SYND_SHUB[ErrorPriority] is the same as the priority of the error in MCA::SHUB::MCA_STATUS_SHUB, then the information in MCA::SHUB::MCA_SYND_SHUB is associated with the error in MCA::SHUB::MCA_DESTAT_SHUB.
52:45	RESERV2. Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	Deferred. 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	RESERV1. Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	AddrLsb. Read-write, Volatile . Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::SHUB::MCA_ADDR_SHUB[ErrorAddr]. A value of 0 indicates that MCA::SHUB::MCA_ADDR_SHUB[63:0] contains a valid byte address. A value of 6 indicates that MCA::SHUB::MCA_ADDR_SHUB[63:6] contains a valid cache line address and that MCA::SHUB::MCA_ADDR_SHUB[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::SHUB::MCA_ADDR_SHUB[63:12] contain a valid 4KB memory page and that MCA::SHUB::MCA_ADDR_SHUB[11:0] should be ignored by error handling software.
23:22	RESERV0. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	ErrorCodeExt. 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	ErrorCode. Read-write, Volatile . Reset: Cold,0000h. Error code for this error.

MSRC000_21D9 [SHUB Deferred Error Address] (MCA::SHUB::MCA_DEADDR_SHUB)Read-write, [Volatile](#). Reset: Cold,0000_0000_0000_0000h.

The MCA::SHUB::MCA_DEADDR_SHUB register stores the address associated with the error in MCA::SHUB::MCA_DESTAT_SHUB. The register is only meaningful if MCA::SHUB::MCA_DESTAT_SHUB[Val]=1 and MCA::SHUB::MCA_DESTAT_SHUB[AddrV]=1. The lowest valid bit of the address is defined by MCA::SHUB::MCA_DESTAT_SHUB[AddrLsb].

_ch1_inst[NBIF[1:0]SHUB0]_n[3:0]_nbio[1:0]_aliasMSR; MSRC000_21D9

Bits	Description
63:0	ErrorAddr. Read-write, Volatile . Reset: Cold,0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::SHUB::MCA_DESTAT_SHUB. The lowest-order valid bit of the address is specified in MCA::SHUB::MCA_DESTAT_SHUB[AddrLsb].

MSRC001_041D [SHUB Machine Check Control Mask] (MCA::SHUB::MCA_CTL_MASK_SHUB)

Read-write. Reset: 0000_0000_0000_0000h.

Inhibit detection of an error source.

_ch1_inst[NBIF[1:0]SHUB0]_n[3:0]_nbio[1:0]_aliasMSR; MSRC001_041D

Bits	Description
63:4	Reserved.
3	SDP_PARITY_ERR. Read-write. Reset: 0. Reset 0. SDP Parity error was detected!
2	NTB_ERR_EVENT. Read-write. Reset: 0. Reset 0. NTB ERROR EVENT was detected!
1	SRAM_ECC_ERR. Read-write. Reset: 0. Reset 0. An SRAM ECC error was detected!
0	NA. Read-write. Reset: 0. Reset 0. no use

MSRC000_21DE [SHUB Machine Check Syndrome Extended] (MCA::SHUB::MCA_SYND1_SHUB)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::SHUB::MCA_STATUS_SHUB [Thread 0](#)

_ch1_inst[NBIF[1:0]SHUB0]_n[3:0]_nbio[1:0]_aliasMSR; MSRC000_21DE

Bits	Description
63:0	Syndrome. Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::SHUB::MCA_SYND1_SHUB register stores information associated with the error in MCA::SHUB::MCA_STATUS_SHUB or MCA_DESTAT. The register is meaningful if MCA::SHUB::MCA_STATUS_SHUB[SyndV]=1. When MCA::SHUB::MCA_CONFIG_SHUB[McaFruTextInMca]=1, MCA::SHUB::MCA_SYND1_SHUB stores ASCII FruText associated with the error.

MSRC000_21DF [SHUB Machine Check Syndrome Extended] (MCA::SHUB::MCA_SYND2_SHUB)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::SHUB::MCA_STATUS_SHUB [Thread 0](#)

_ch1_inst[NBIF[1:0]SHUB0]_n[3:0]_nbio[1:0]_aliasMSR; MSRC000_21DF

Bits	Description
63:0	Syndrome. Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::SHUB::MCA_SYND2_SHUB register stores information associated with the error in MCA::SHUB::MCA_STATUS_SHUB or MCA_DESTAT. The register is meaningful if MCA::SHUB::MCA_STATUS_SHUB[SyndV]=1. When MCA::SHUB::MCA_CONFIG_SHUB[McaFruTextInMca]=1, MCA::SHUB::MCA_SYND2_SHUB stores ASCII FruText associated with the error.

3.2.5.22 USB**MSR0000_0468...MSRC000_21A0 [USB Machine Check Control] (MCA::USB::MCA_CTL_USB)**

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::USB::MCA_CTL_USB register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG_CTL. Does not affect error detection, correction, or logging.

_inst[USB[1:0]USB]_n[1:0]_aliasMSRLEGACY; MSR0000_0468

_inst[USB[1:0]USB]_n[1:0]_aliasMSR; MSRC000_21A0

Bits	Description
63:6	Reserved.
5	AXI_slv. Read-write. Reset: 0. axi slave response error
4	PHY_RAM1. Read-write. Reset: 0. parity error for phy ram1
3	PHY_RAM0. Read-write. Reset: 0. parity error for phy ram0
2	S0_RAM2. Read-write. Reset: 0. ecc error for s0 ram2
1	S0_RAM1. Read-write. Reset: 0. ecc error for s0 ram1
0	S0_RAM0. Read-write. Reset: 0. ecc error for s0 ram0

MSR0000_0469...MSRC000_21A1 [USB Machine Check Status] (MCA::USB::MCA_STATUS_USB)

Reset: Cold,0000_0000_0000_0000h.

Logs information associated with errors.

_inst[USB[1:0]USB]_n[1:0]_aliasMSRLEGACY; MSR0000_0469

_inst[USB[1:0]USB]_n[1:0]_aliasMSR; MSRC000_21A1

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::USB::MCA_CTL_USB. This bit is a copy of bit in MCA::USB::MCA_CTL_USB for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	MiscV. Reset: Cold,0. 1=Valid thresholding in MCA::USB::MCA_MISC0_USB. 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	AddrV. Reset: Cold,0. 1=MCA::USB::MCA_ADDR_USB contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	PCC. 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	ErrCoreIdVal. 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	TCC. 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::USB::MCA_STATUS_USB[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	RESERV54. 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	SyndV. Reset: Cold,0. 1=This error logged information in MCA::USB::MCA_SYND_USB. If MCA::USB::MCA_SYND_USB[ErrorPriority] is the same as the priority of the error in MCA::USB::MCA_STATUS_USB, then the information in MCA::USB::MCA_SYND_USB is associated with the error in MCA::USB::MCA_STATUS_USB. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::USB::MCA_ADDR_USB[ErrorAddr]. A value of 0 indicates that MCA::USB::MCA_ADDR_USB[63:0] contains a valid byte address. A value of 6 indicates that MCA::USB::MCA_ADDR_USB[63:6] contains a valid cache line address and that MCA::USB::MCA_ADDR_USB[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::USB::MCA_ADDR_USB[63:12] contain a valid 4KB memory page and that MCA::USB::MCA_ADDR_USB[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	RESERV22. 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	ErrorCodeExt. 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::USB::MCA_CTL_USB 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	ErrorCode. 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 115: MCA_STATUS_USB

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
S0_RAM0	0x0	0/1	0/1	0/1	0	0/1	0
S0_RAM1	0x1	0/1	0/1	0/1	0	0/1	0
S0_RAM2	0x2	0/1	0/1	0/1	0	0/1	0
PHY_RAM0	0x3	1	1	1	0	0	0
PHY_RAM1	0x4	1	1	1	0	0	0
AXI_slv	0x5	0	0	0	0	0	0

MSR0000_046A...MSRC000_21A2 [USB Machine Check Address] (MCA::USB::MCA_ADDR_USB)

Read-only. Reset: Cold,0000_0000_0000_0000h.

MCA::USB::MCA_ADDR_USB stores an address and other information associated with the error in MCA::USB::MCA_STATUS_USB. The register is only meaningful if MCA::USB::MCA_STATUS_USB[Val]=1 and MCA::USB::MCA_STATUS_USB[AddrV]=1.

_inst[USB[1:0]USB]_n[1:0]_aliasMSRLEGACY; MSR0000_046A

_inst[USB[1:0]USB]_n[1:0]_aliasMSR; MSRC000_21A2

Bits	Description
63:0	ErrorAddr. Read-only. Reset: Cold,0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::USB::MCA_STATUS_USB.

Table 116: MCA_ADDR_USB

Error Type	Bits	Description
S0_RAM0	[63:0]	Error address for S0 RAM0
S0_RAM1	[63:0]	Error address for S0 RAM1
S0_RAM2	[63:0]	Error address for S0 RAM2
PHY_RAM0	[63:0]	Error address for PHY RAM0
PHY_RAM1	[63:0]	Error address for PHY RAM1
AXI_slv	[63:0]	Error address for AXI Slave

MSR0000_046B...MSRC000_21A3 [USB Machine Check Miscellaneous 0] (MCA::USB::MCA_MISC0_USB)

Log miscellaneous information associated with errors.

_inst[USB[1:0]USB]_n[1:0]_aliasMSRLEGACY; MSR0000_046B

_inst[USB[1:0]USB]_n[1:0]_aliasMSR; MSRC000_21A3

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr:: HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr:: HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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::USB::MCA_MISC0_USB[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	LvtOffset. 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::USB::MCA_MISC0_USB[Locked]) ? Read-write : Read-only.
51	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr:: HWCR[McStatusWrEn] !MCA::USB::MCA_MISC0_USB[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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::USB::MCA_MISC0_USB[Locked]) ? Read-write : Read-only.
48	Ovrflw. 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::USB::MCA_MISC0_USB[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	ErrCnt. 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::USB::MCA_MISC0_USB[Locked]) ? Read-write : Read-only.
31:24	BlkPtr. Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

MSRC000_21A4 [USB Machine Check Configuration] (MCA::USB::MCA_CONFIG_USB)

Reset: 0000_0000_0000_0121h.

Controls configuration of the associated machine check bank.

_inst[USB[1:0]USB]_n[1:0]_aliasMSR; MSRC000_21A4

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	McaXEnable. 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	IntPresent. Read-only, Volatile . Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::USB::MCA_CONFIG_USB[IntEn].
9	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::USB::MCA_CONFIG_USB[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. Read-only. Reset: 1. 1=MCA::USB::MCA_CONFIG_USB[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::USB::MCA_CONFIG_USB[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	DeferredErrorLoggingSupported. 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	McaX. Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::USB::MCA_MISC0_USB[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::USB::MCA_STATUS_USB[TCC] is present.

MSRC000_21A5 [USB IP Identification] (MCA::USB::MCA_IPID_USB)

Reset: 0000_00AA_0000_0000h.

The MCA::USB::MCA_IPID_USB register is used by software to determine what IP type and revision is associated with the MCA bank.

_inst[USB[1:0]USB]_n[1:0]_aliasMSR; MSRC000_21A5

Bits	Description
63:48	McaType. Read-only. Reset: 0000h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi. 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	HardwareID. Read-only. Reset: 0AAh. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId. Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _instUSB0USB_n0_aliasMSR: 16DF_E000h
	Init: _instUSB1USB_n1_aliasMSR: 16FF_E000h

MSRC000_21A6 [USB Machine Check Syndrome] (MCA::USB::MCA_SYND_USB)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::USB::MCA_STATUS_USB [Thread 0](#)

_inst[USB[1:0]USB]_n[1:0]_aliasMSR; MSRC000_21A6

Bits	Description
63:32	Syndrone. Read-write, Volatile . Reset: Cold, 0000_0000h. Contains the syndrome, if any, associated with the error logged in MCA::USB::MCA_STATUS_USB. The low-order bit of the syndrome is stored in bit 0, and the syndrome has a length specified by MCA::USB::MCA_SYND_USB[Length]. The Syndrome field is only valid when MCA::USB::MCA_SYND_USB[Length] is not 0.
31:27	Reserved.
26:24	ErrorPriority. Read-write, Volatile . Reset: Cold, 0h. Encodes the priority of the error logged in MCA::USB::MCA_SYND_USB. 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	Length. Read-write, Volatile . Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::USB::MCA_SYND_USB[Syndrome]. Length values greater than 32 (decimal) are interpreted as equal to 32 (decimal). A value of 0 indicates that there is no valid syndrome in MCA::USB::MCA_SYND_USB. For example, a syndrome length of 9 means that MCA::USB::MCA_SYND_USB[Syndrome] bits [8:0] contains a valid syndrome.
17:0	ErrorInformation. Read-write, Volatile . Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 117 [MCA_SYND_USB].

Table 117: MCA_SYND_USB

Error Type	Bits	Description
S0_RAM0	[17:0]	Reserved
S0_RAM1	[17:0]	Reserved
S0_RAM2	[17:0]	Reserved
PHY_RAM0	[17:0]	Reserved
PHY_RAM1	[17:0]	Reserved
AXI_slv	[17:0]	Reserved

MSRC001_041A [USB Machine Check Control Mask] (MCA::USB::MCA_CTL_MASK_USB)

Read-write. Reset: 0000_0000_0000_0000h.

Inhibit detection of an error source.

_inst[USB[1:0]USB]_n[1:0]_aliasMSR; MSRC001_041A

Bits	Description
63:6	Reserved.
5	AXI_slv. Read-write. Reset: 0. axi slave response error
4	PHY_RAM1. Read-write. Reset: 0. parity error for phy ram1
3	PHY_RAM0. Read-write. Reset: 0. parity error for phy ram0
2	S0_RAM2. Read-write. Reset: 0. ecc error for s0 ram2
1	S0_RAM1. Read-write. Reset: 0. ecc error for s0 ram1
0	S0_RAM0. Read-write. Reset: 0. ecc error for s0 ram0

MSRC000_21AE [USB Machine Check Syndrome Extended] (MCA::USB::MCA_SYND1_USB)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::USB::MCA_STATUS_USB [Thread 0](#)[_inst\[USB\[1:0\]USB\]_n\[1:0\]_aliasMSR; MSRC000_21AE](#)

Bits	Description
63:0	Syndrom . Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::USB::MCA_SYND1_USB register stores information associated with the error in MCA::USB::MCA_STATUS_USB or MCA_DESTAT. The register is meaningful if MCA::USB::MCA_STATUS_USB[SyndV]=1. When MCA::USB::MCA_CONFIG_USB[McaFruTextInMca]=1, MCA::USB::MCA_SYND1_USB stores ASCII FruText associated with the error.

MSRC000_21AF [USB Machine Check Syndrome Extended] (MCA::USB::MCA_SYND2_USB)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::USB::MCA_STATUS_USB [Thread 0](#)[_inst\[USB\[1:0\]USB\]_n\[1:0\]_aliasMSR; MSRC000_21AF](#)

Bits	Description
63:0	Syndrom . Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::USB::MCA_SYND2_USB register stores information associated with the error in MCA::USB::MCA_STATUS_USB or MCA_DESTAT. The register is meaningful if MCA::USB::MCA_STATUS_USB[SyndV]=1. When MCA::USB::MCA_CONFIG_USB[McaFruTextInMca]=1, MCA::USB::MCA_SYND2_USB stores ASCII FruText associated with the error.

3.2.5.23 SATA**MSR0000_0468...MSRC000_21A0 [SATA Machine Check Control] (MCA::SATA::MCA_CTL_SATA)**

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::SATA::MCA_CTL_SATA register must be enabled by the corresponding enable bit in Core::X86::Msr::[MCG_CTL](#). Does not affect error detection, correction, or logging.

[_inst\[SATA\[3:0\]SATA\]_n\[3:0\]_aliasMSRLEGACY; MSR0000_0468](#)[_inst\[SATA\[3:0\]SATA\]_n\[3:0\]_aliasMSR; MSRC000_21A0](#)

Bits	Description
63:8	Reserved.
7	perr_p7 . Read-write. Reset: 0. parity error for port 7
6	perr_p6 . Read-write. Reset: 0. parity error for port 6
5	perr_p5 . Read-write. Reset: 0. parity error for port 5
4	perr_p4 . Read-write. Reset: 0. parity error for port 4
3	perr_p3 . Read-write. Reset: 0. parity error for port 3
2	perr_p2 . Read-write. Reset: 0. parity error for port 2
1	perr_p1 . Read-write. Reset: 0. parity error for port 1
0	perr_p0 . Read-write. Reset: 0. parity error for port 0

MSR0000_0469...MSRC000_21A1 [SATA Machine Check Status] (MCA::SATA::MCA_STATUS_SATA)

Reset: Cold,0000_0000_0000_0000h.

Logs information associated with errors.

_inst[SATA[3:0]SATA]_n[3:0]_aliasMSRLEGACY; MSR0000_0469

_inst[SATA[3:0]SATA]_n[3:0]_aliasMSR; MSRC000_21A1

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::SATA::MCA_CTL_SATA. This bit is a copy of bit in MCA::SATA::MCA_CTL_SATA for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	MiscV. Reset: Cold,0. 1=Valid thresholding in MCA::SATA::MCA_MISC0_SATA. 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	AddrV. Reset: Cold,0. 1=MCA::SATA::MCA_ADDR_SATA contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	PCC. 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	ErrCoreIdVal. 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	TCC. 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::SATA::MCA_STATUS_SATA[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	RESERV54. 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	SyndV. Reset: Cold,0. 1=This error logged information in MCA::SATA::MCA_SYND_SATA. If MCA::SATA::MCA_SYND_SATA[ErrorPriority] is the same as the priority of the error in MCA::SATA::MCA_STATUS_SATA, then the information in MCA::SATA::MCA_SYND_SATA is associated with the error in MCA::SATA::MCA_STATUS_SATA. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::SATA::MCA_ADDR_SATA[ErrorAddr]. A value of 0 indicates that MCA::SATA::MCA_ADDR_SATA[63:0] contains a valid byte address. A value of 6 indicates that MCA::SATA::MCA_ADDR_SATA[63:6] contains a valid cache line address and that MCA::SATA::MCA_ADDR_SATA[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::SATA::MCA_ADDR_SATA[63:12] contain a valid 4KB memory page and that MCA::SATA::MCA_ADDR_SATA[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	RESERV22. 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	ErrorCodeExt. 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::SATA::MCA_CTL_SATA 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	ErrorCode. 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 118: MCA_STATUS_SATA

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
perr_p0	0x0	1	1	1	0	0	0
perr_p1	0x1	1	1	1	0	0	0
perr_p2	0x2	1	1	1	0	0	0
perr_p3	0x3	1	1	1	0	0	0
perr_p4	0x4	1	1	1	0	0	0
perr_p5	0x5	1	1	1	0	0	0
perr_p6	0x6	1	1	1	0	0	0
perr_p7	0x7	1	1	1	0	0	0

MSR0000_046A...MSRC000_21A2 [SATA Machine Check Address] (MCA::SATA::MCA_ADDR_SATA)

Read-only. Reset: Cold,0000_0000_0000_0000h.

MCA::SATA::MCA_ADDR_SATA stores an address and other information associated with the error in MCA::SATA::MCA_STATUS_SATA. The register is only meaningful if MCA::SATA::MCA_STATUS_SATA[Val]=1 and MCA::SATA::MCA_STATUS_SATA[AddrV]=1.

_inst[SATA[3:0]SATA]_n[3:0]_aliasMSRLEGACY; MSR0000_046A

_inst[SATA[3:0]SATA]_n[3:0]_aliasMSR; MSRC000_21A2

Bits	Description
63:0	ErrorAddr. Read-only. Reset: Cold,0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::SATA::MCA_STATUS_SATA.

Table 119: MCA_ADDR_SATA

Error Type	Bits	Description
perr_p0	[63:0]	Reserved
perr_p1	[63:0]	Reserved
perr_p2	[63:0]	Reserved
perr_p3	[63:0]	Reserved
perr_p4	[63:0]	Reserved
perr_p5	[63:0]	Reserved
perr_p6	[63:0]	Reserved
perr_p7	[63:0]	Reserved

MSR0000_046B...MSRC000_21A3 [SATA Machine Check Miscellaneous 0] (MCA::SATA::MCA_MISC0_SATA)

Log miscellaneous information associated with errors.

_inst[SATA[3:0]SATA]_n[3:0]_aliasMSRLEGACY; MSR0000_046B

_inst[SATA[3:0]SATA]_n[3:0]_aliasMSR; MSRC000_21A3

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr:: HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr:: HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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::SATA::MCA_MISC0_SATA[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	LvtOffset. 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::SATA::MCA_MISC0_SATA[Locked]) ? Read-write : Read-only.
51	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr:: HWCR[McStatusWrEn] !MCA::SATA::MCA_MISC0_SATA[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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::SATA::MCA_MISC0_SATA[Locked]) ? Read-write : Read-only.
48	Ovrflw. 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::SATA::MCA_MISC0_SATA[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	ErrCnt. 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::SATA::MCA_MISC0_SATA[Locked]) ? Read-write : Read-only.
31:24	BlkPtr. Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

MSRC000_21A4 [SATA Machine Check Configuration] (MCA::SATA::MCA_CONFIG_SATA)

Reset: 0000_0002_0000_0125h.

Controls configuration of the associated machine check bank.

_inst[SATA[3:0]SATA]_n[3:0]_aliasMSR; MSRC000_21A4

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	LogDeferredInMcaStat. Read-write. Reset: 0. Init: BIOS,1. 1=Log deferred errors in MCA::SATA::MCA_STATUS_SATA and MCA::SATA::MCA_ADDR_SATA in addition to MCA::SATA::MCA_DESTAT_SATA and MCA::SATA::MCA_DEADDR_SATA. 0=Only log deferred errors in MCA::SATA::MCA_DESTAT_SATA and MCA::SATA::MCA_DEADDR_SATA. This bit does not affect logging of deferred errors in MCA::SATA::MCA_SYND_SATA, MCA::SATA::MCA_MISC0_SATA.
33	Reserved.
32	McaXEnable. 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	IntPresent. Read-only, Volatile . Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::SATA::MCA_CONFIG_SATA[IntEn].
9	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::SATA::MCA_CONFIG_SATA[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. Read-only. Reset: 1. 1=MCA::SATA::MCA_CONFIG_SATA[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::SATA::MCA_CONFIG_SATA[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	DeferredErrorLoggingSupported. Read-only. Reset: 1. 1=Deferred errors are supported in this MCA bank, and MCA::SATA::MCA_CONFIG_SATA[LogDeferredInMcaStat] controls the logging behavior of these errors. MCA::SATA::MCA_DESTAT_SATA and MCA::SATA::MCA_DEADDR_SATA are supported in this MCA bank. 0=Deferred errors are not supported in this bank.
1	Reserved.
0	McaX. Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::SATA::MCA_MISC0_SATA[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::SATA::MCA_STATUS_SATA[TCC] is present.

MSRC000_21A5 [SATA IP Identification] (MCA::SATA::MCA_IPID_SATA)

Reset: 0000_00A8_0000_0000h.

The MCA::SATA::MCA_IPID_SATA register is used by software to determine what IP type and revision is associated with the MCA bank.

_inst[SATA[3:0]SATA]_n[3:0]_aliasMSR; MSRC000_21A5

Bits	Description
63:48	McaType. Read-only. Reset: 0000h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi. 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	HardwareID. Read-only. Reset: 0A8h. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId. Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _instSATA0SATA_n0_aliasMSR: 0310_2000h
	Init: _instSATA1SATA_n1_aliasMSR: 0320_2000h
	Init: _instSATA2SATA_n2_aliasMSR: 0330_2000h
	Init: _instSATA3SATA_n3_aliasMSR: 0340_2000h

MSRC000_21A6 [SATA Machine Check Syndrome] (MCA::SATA::MCA_SYND_SATA)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::SATA::MCA_STATUS_SATA [Thread 0](#)

_inst[SATA[3:0]SATA]_n[3:0]_aliasMSR; MSRC000_21A6

Bits	Description
63:32	Syndrom. Read-write, Volatile . Reset: Cold, 0000_0000h. Contains the syndrome, if any, associated with the error logged in MCA::SATA::MCA_STATUS_SATA. The low-order bit of the syndrome is stored in bit 0, and the syndrome has a length specified by MCA::SATA::MCA_SYND_SATA[Length]. The Syndrome field is only valid when MCA::SATA::MCA_SYND_SATA[Length] is not 0.
31:27	Reserved.
26:24	ErrorPriority. Read-write, Volatile . Reset: Cold, 0h. Encodes the priority of the error logged in MCA::SATA::MCA_SYND_SATA. 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	Length. Read-write, Volatile . Reset: Cold, 00h. Specifies the length in bits of the syndrome contained in MCA::SATA::MCA_SYND_SATA[Syndrome]. Length values greater than 32 (decimal) are interpreted as equal to 32 (decimal). A value of 0 indicates that there is no valid syndrome in MCA::SATA::MCA_SYND_SATA. For example, a syndrome length of 9 means that MCA::SATA::MCA_SYND_SATA[Syndrome] bits [8:0] contains a valid syndrome.
17:0	ErrorInformation. Read-write, Volatile . Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 120 [MCA_SYND_SATA].

Table 120: MCA_SYND_SATA

Error Type	Bits	Description
perr_p0	[17:3]	Reserved
	[2]	H2D memory parity error
	[1]	D2H memory parity error
	[0]	Context memory parity error
perr_p1	[17:3]	Reserved
	[2]	H2D memory parity error
	[1]	D2H memory parity error
	[0]	Context memory parity error
perr_p2	[17:3]	Reserved
	[2]	H2D memory parity error
	[1]	D2H memory parity error

	[0]	Context memory parity error
perr_p3	[17:3]	Reserved
	[2]	H2D memory parity error
	[1]	D2H memory parity error
	[0]	Context memory parity error
perr_p4	[17:3]	Reserved
	[2]	H2D memory parity error
	[1]	D2H memory parity error
	[0]	Context memory parity error
perr_p5	[17:3]	Reserved
	[2]	H2D memory parity error
	[1]	D2H memory parity error
	[0]	Context memory parity error
perr_p6	[17:3]	Reserved
	[2]	H2D memory parity error
	[1]	D2H memory parity error
	[0]	Context memory parity error
perr_p7	[17:3]	Reserved
	[2]	H2D memory parity error
	[1]	D2H memory parity error
	[0]	Context memory parity error

MSRC000_21A8 [SATA Machine Check Deferred Error Status] (MCA::SATA::MCA_DESTAT_SATA)

Reset: Cold,0000_0000_0000_0000h.

Holds status information for the first deferred error seen in this bank.

_inst[SATA[3:0]SATA]_n[3:0]_aliasMSR; MSRC000_21A8

Bits	Description
63	Val. Read-write, Volatile . Reset: Cold,0. 1=A valid error has been detected (whether it is enabled or not).
62	Overflow. 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 overwrite priorities.)
61:59	RESERV4. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
58	AddrV. Read-write, Volatile . Reset: Cold,0. 1=MCA::SATA::MCA_DEADDR_SATA contains address information associated with the error.
57:54	RESERV3. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
53	SyndV. Read-write, Volatile . Reset: Cold,0. 1=This error logged information in MCA::SATA::MCA_SYND_SATA. If MCA::SATA::MCA_SYND_SATA[ErrorPriority] is the same as the priority of the error in MCA::SATA::MCA_STATUS_SATA, then the information in MCA::SATA::MCA_SYND_SATA is associated with the error in MCA::SATA::MCA_DESTAT_SATA.
52:45	RESERV2. Read-write. Reset: Cold,00h. MCA_DEFSTAT Register Reserved bits.
44	Deferred. 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	RESERV1. Read-write. Reset: Cold,0000h. MCA_DEFSTAT Register Reserved bits.
29:24	AddrLsb. Read-write, Volatile . Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::SATA::MCA_ADDR_SATA[ErrorAddr]. A value of 0 indicates that MCA::SATA::MCA_ADDR_SATA[63:0] contains a valid byte address. A value of 6 indicates that MCA::SATA::MCA_ADDR_SATA[63:6] contains a valid cache line address and that MCA::SATA::MCA_ADDR_SATA[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::SATA::MCA_ADDR_SATA[63:12] contain a valid 4KB memory page and that MCA::SATA::MCA_ADDR_SATA[11:0] should be ignored by error handling software.
23:22	RESERV0. Read-write. Reset: Cold,0h. MCA_DEFSTAT Register Reserved bits.
21:16	ErrorCodeExt. 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	ErrorCode. Read-write, Volatile . Reset: Cold,0000h. Error code for this error.

MSRC000_21A9 [SATA Deferred Error Address] (MCA::SATA::MCA_DEADDR_SATA)

Read-only. Reset: Cold,0000_0000_0000_0000h.

The MCA::SATA::MCA_DEADDR_SATA register stores the address associated with the error in MCA::SATA::MCA_DESTAT_SATA. The register is only meaningful if MCA::SATA::MCA_DESTAT_SATA[Val]=1 and MCA::SATA::MCA_DESTAT_SATA[AddrV]=1. The lowest valid bit of the address is defined by MCA::SATA::MCA_DESTAT_SATA[AddrLsb].

_inst[SATA[3:0]SATA]_n[3:0]_aliasMSR; MSRC000_21A9

Bits	Description
63:0	ErrorAddr. Read-only. Reset: Cold,0000_0000_0000_0000h. Contains the address, if any, associated with the error logged in MCA::SATA::MCA_DESTAT_SATA.

MSRC001_041A [SATA Machine Check Control Mask] (MCA::SATA::MCA_CTL_MASK_SATA)

Read-write. Reset: 0000_0000_0000_0000h.

Inhibit detection of an error source.

_inst[SATA[3:0]SATA]_n[3:0]_aliasMSR; MSRC001_041A

Bits	Description
63:8	Reserved.
7	perr_p7 . Read-write. Reset: 0. parity error for port 7
6	perr_p6 . Read-write. Reset: 0. parity error for port 6
5	perr_p5 . Read-write. Reset: 0. parity error for port 5
4	perr_p4 . Read-write. Reset: 0. parity error for port 4
3	perr_p3 . Read-write. Reset: 0. parity error for port 3
2	perr_p2 . Read-write. Reset: 0. parity error for port 2
1	perr_p1 . Read-write. Reset: 0. parity error for port 1
0	perr_p0 . Read-write. Reset: 0. parity error for port 0

MSRC000_21AE [SATA Machine Check Syndrome Extended] (MCA::SATA::MCA_SYND1_SATA)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::SATA::MCA_STATUS_SATA [Thread 0](#)

_inst[SATA[3:0]SATA]_n[3:0]_aliasMSR; MSRC000_21AE

Bits	Description
63:0	Syndrome . Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::SATA::MCA_SYND1_SATA register stores information associated with the error in MCA::SATA::MCA_STATUS_SATA or MCA_DESTAT. The register is meaningful if MCA::SATA::MCA_STATUS_SATA[SyndV]=1. When MCA::SATA::MCA_CONFIG_SATA[McaFruTextInMca]=1, MCA::SATA::MCA_SYND1_SATA stores ASCII FruText associated with the error.

MSRC000_21AF [SATA Machine Check Syndrome Extended] (MCA::SATA::MCA_SYND2_SATA)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::SATA::MCA_STATUS_SATA [Thread 0](#)

_inst[SATA[3:0]SATA]_n[3:0]_aliasMSR; MSRC000_21AF

Bits	Description
63:0	Syndrome . Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::SATA::MCA_SYND2_SATA register stores information associated with the error in MCA::SATA::MCA_STATUS_SATA or MCA_DESTAT. The register is meaningful if MCA::SATA::MCA_STATUS_SATA[SyndV]=1. When MCA::SATA::MCA_CONFIG_SATA[McaFruTextInMca]=1, MCA::SATA::MCA_SYND2_SATA stores ASCII FruText associated with the error.

3.2.5.24 MPDMA

MSR0000_0450...MSRC000_2140 [MPDMA Machine Check Control] (MCA::MPDMA::MCA_CTL_MPDMA)

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::MPDMA::MCA_CTL_MPDMA register must be enabled by the corresponding enable bit in Core::X86::Msr::MCG_CTL. Does not affect error detection, correction, or logging.

_instMPDMAMPDMA1_n[5:0]_aliasMSRLEGACY; MSR0000_0450

_instMPDMAMPDMA1_n[5:0]_aliasMSR; MSRC000_2140

Bits	Description
63:49	Reserved.
48	MpDMAPteCmdmemInternalFifoMemError . Read-write. Reset: 0. MPDMA PTE Command Memory Internal ECC or parity error
47	MpDMAPteCmdmemDMAFifoMemError . Read-write. Reset: 0. MPDMA PTE Command Memory DMA ECC or parity error
46	MpDMAPteInternalDataFifoMemError . Read-write. Reset: 0. MPDMA PTE Internal Data FIFO ECC or parity error
45	MpDMAPteHubDataFifoMemError . Read-write. Reset: 0. MPDMA PTE Hub Data FIFO ECC or parity error
44	MpDMAPteCmdfifoMemError . Read-write. Reset: 0. MPDMA PTE Command FIFO ECC or parity error
43	sdp_watchdog_timeout_error . Read-write. Reset: 0. sdp watchdog timer expired
42:29	Reserved.
28	MpDMASHubIfRdBufError . Read-write. Reset: 0. System Hub Read Buffer ECC or parity error.
27	MpDMAITagBError_2 . Read-write. Reset: 0. Instruction Tag Cache Bank B ECC or parity error.
26	MpDMAITagAError_2 . Read-write. Reset: 0. Instruction Tag Cache Bank A ECC or parity error.
25	MpDMAICacheBError_2 . Read-write. Reset: 0. Instruction Cache Bank B ECC or parity error.
24	MpDMAICacheAError_2 . Read-write. Reset: 0. Instruction Cache Bank A ECC or parity error.
23	MpDMADTagBError_2 . Read-write. Reset: 0. Data Tag Cache Bank B ECC or parity error.
22	MpDMADTagAError_2 . Read-write. Reset: 0. Data Tag Cache Bank A ECC or parity error.
21	MpDMADCacheBError_2 . Read-write. Reset: 0. Data Cache Bank B ECC or parity error.
20	MpDMADCacheAError_2 . Read-write. Reset: 0. Data Cache Bank A ECC or parity error.
19	MpDMAITagBError_1 . Read-write. Reset: 0. Instruction Tag Cache Bank B ECC or parity error.
18	MpDMAITagAError_1 . Read-write. Reset: 0. Instruction Tag Cache Bank A ECC or parity error.
17	MpDMAICacheBError_1 . Read-write. Reset: 0. Instruction Cache Bank B ECC or parity error.
16	MpDMAICacheAError_1 . Read-write. Reset: 0. Instruction Cache Bank A ECC or parity error.
15	MpDMADTagBError_1 . Read-write. Reset: 0. Data Tag Cache Bank B ECC or parity error.
14	MpDMADTagAError_1 . Read-write. Reset: 0. Data Tag Cache Bank A ECC or parity error.
13	MpDMADCacheBError_1 . Read-write. Reset: 0. Data Cache Bank B ECC or parity error.
12	MpDMADCacheAError_1 . Read-write. Reset: 0. Data Cache Bank A ECC or parity error.
11	MpDMAITagBError . Read-write. Reset: 0. Instruction Tag Cache Bank B ECC or parity error.
10	MpDMAITagAError . Read-write. Reset: 0. Instruction Tag Cache Bank A ECC or parity error.
9	MpDMAICacheBError . Read-write. Reset: 0. Instruction Cache Bank B ECC or parity error.
8	MpDMAICacheAError . Read-write. Reset: 0. Instruction Cache Bank A ECC or parity error.
7	MpDMADTagBError . Read-write. Reset: 0. Data Tag Cache Bank B ECC or parity error.
6	MpDMADTagAError . Read-write. Reset: 0. Data Tag Cache Bank A ECC or parity error.
5	MpDMADCacheBError . Read-write. Reset: 0. Data Cache Bank B ECC or parity error.
4	MpDMADCacheAError . Read-write. Reset: 0. Data Cache Bank A ECC or parity error.
3	MpDMASram3Error . Read-write. Reset: 0. Main SRAM [127:96] bank ECC or parity error.
2	MpDMASram2Error . Read-write. Reset: 0. Main SRAM [95:64] bank ECC or parity error.
1	MpDMASram1Error . Read-write. Reset: 0. Main SRAM [63:32] bank ECC or parity error.
0	MpDMASram0Error . Read-write. Reset: 0. Main SRAM [31:0] bank ECC or parity error.

**MSR0000_0451...MSRC000_2141 [MPDMA Machine Check Status]
(MCA::MPDMA::MCA_STATUS_MPDMA)**

Reset: Cold,0000_0000_0000_0000h.

Logs information associated with errors.

_instMPDMAMPDMA1_n[5:0]_aliasMSRLEGACY; MSR0000_0451

_instMPDMAMPDMA1_n[5:0]_aliasMSR; MSRC000_2141

Bits	Description
63	Val. 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	Overflow. 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	UC. 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	En. Reset: Cold,0. 1=MCA error reporting is enabled for this error, as indicated by the corresponding bit in MCA::MPDMA::MCA_CTL_MPDMA. This bit is a copy of bit in MCA::MPDMA::MCA_CTL_MPDMA for this error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
59	MiscV. Reset: Cold,0. 1=Valid thresholding in MCA::MPDMA::MCA_MISC0_MPDMA. 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	AddrV. Reset: Cold,0. 1=MCA::MPDMA::MCA_ADDR_MPDMA contains address information associated with the error. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
57	PCC. 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	ErrCoreIdVal. 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	TCC. 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::MPDMA::MCA_STATUS_MPDMA[PCC]=0. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
54	RESERV54. 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	SyndV. Reset: Cold,0. 1=This error logged information in MCA::MPDMA::MCA_SYND_MPDMA. If MCA::MPDMA::MCA_SYND_MPDMA[ErrorPriority] is the same as the priority of the error in MCA::MPDMA::MCA_STATUS_MPDMA, then the information in MCA::MPDMA::MCA_SYND_MPDMA is associated with the error in MCA::MPDMA::MCA_STATUS_MPDMA. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read,Write-0-only,Error-on-write-1.
52	Reserved.
51:47	RESERV47. 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	CECC. 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	UECC. 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	Deferred. 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	Poison. 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	RESERV41. 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	Scrub. 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	RESERV38. 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	ErrCoreId. 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	RESERV30. 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	AddrLsb. Reset: Cold,00h. Specifies the least significant valid bit of the address contained in MCA::MPDMA::MCA_ADDR_MPDMA[ErrorAddr]. A value of 0 indicates that MCA::MPDMA::MCA_ADDR_MPDMA[63:0] contains a valid byte address. A value of 6 indicates that MCA::MPDMA::MCA_ADDR_MPDMA[63:6] contains a valid cache line address and that MCA::MPDMA::MCA_ADDR_MPDMA[5:0] are not part of the address and should be ignored by error handling software. A value of 12 indicates that MCA::MPDMA::MCA_ADDR_MPDMA[63:12] contain a valid 4KB memory page and that MCA::MPDMA::MCA_ADDR_MPDMA[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	RESERV22. 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	ErrorCodeExt. 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::MPDMA::MCA_CTL_MPDMA 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	ErrorCode. 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 121: MCA_STATUS_MPDMA

Error Type	ErrorCode Ext	UC	PCC	TCC	Deferred	Poison	AddrV
MpDMASra m0Error	0x0	0/1	0/1	0/1	0	0	1
MpDMASra m1Error	0x1	0/1	0/1	0/1	0	0	1
MpDMASra m2Error	0x2	0/1	0/1	0/1	0	0	1
MpDMASra m3Error	0x3	0/1	0/1	0/1	0	0	1

MpDMADC acheAError	0x4	0/1	0/1	0/1	0	0	1
MpDMADC acheBError	0x5	0/1	0/1	0/1	0	0	1
MpDMADT agAError	0x6	0/1	0/1	0/1	0	0	1
MpDMADT agBError	0x7	0/1	0/1	0/1	0	0	1
MpDMAICa cheAError	0x8	0/1	0/1	0/1	0	0	1
MpDMAICa cheBError	0x9	0/1	0/1	0/1	0	0	1
MpDMAITa gAError	0xa	0/1	0/1	0/1	0	0	1
MpDMAITa gBError	0xb	0/1	0/1	0/1	0	0	1
MpDMADC acheAError_ 1	0xc	0/1	0/1	0/1	0	0	1
MpDMADC acheBError_ 1	0xd	0/1	0/1	0/1	0	0	1
MpDMADT agAError_1	0xe	0/1	0/1	0/1	0	0	1
MpDMADT agBError_1	0xf	0/1	0/1	0/1	0	0	1
MpDMAICa cheAError_1	0x10	0/1	0/1	0/1	0	0	1
MpDMAICa cheBError_1	0x11	0/1	0/1	0/1	0	0	1
MpDMAITa gAError_1	0x12	0/1	0/1	0/1	0	0	1
MpDMAITa gBError_1	0x13	0/1	0/1	0/1	0	0	1
MpDMADC acheAError_ 2	0x14	0/1	0/1	0/1	0	0	1
MpDMADC acheBError_ 2	0x15	0/1	0/1	0/1	0	0	1
MpDMADT agAError_2	0x16	0/1	0/1	0/1	0	0	1
MpDMADT agBError_2	0x17	0/1	0/1	0/1	0	0	1
MpDMAICa cheAError_2	0x18	0/1	0/1	0/1	0	0	1
MpDMAICa cheBError_2	0x19	0/1	0/1	0/1	0	0	1
MpDMAITa gAError_2	0x1a	0/1	0/1	0/1	0	0	1
MpDMAITa	0x1b	0/1	0/1	0/1	0	0	1

gBError_2							
MpDMASHubIfRdBufError	0x1c	0/1	0/1	0/1	0	0	1
MpDMATvfDVSecError	0x1d	0/1	0/1	0/1	0	0	1
MpDMATvfMMioMbox0Error	0x1e	0/1	0/1	0/1	0	0	1
MpDMATvfMmioMbox1Error	0x1f	0/1	0/1	0/1	0	0	1
MpDMATvfDoorbellMemError	0x20	0/1	0/1	0/1	0	0	1
MpDMATvfSdpSlaveMem0Error	0x21	0/1	0/1	0/1	0	0	1
MpDMATvfSdpSlaveMem1Error	0x22	0/1	0/1	0/1	0	0	1
MpDMATvfSdpSlaveMem2Error	0x23	0/1	0/1	0/1	0	0	1
MpDMATvfSdpMasterMem0Error	0x24	0/1	0/1	0/1	0	0	1
MpDMATvfSdpMasterMem1Error	0x25	0/1	0/1	0/1	0	0	1
MpDMATvfSdpMasterMem2Error	0x26	0/1	0/1	0/1	0	0	1
MpDMATvfSdpMasterMem3Error	0x27	0/1	0/1	0/1	0	0	1
MpDMATvfSdpMasterMem4Error	0x28	0/1	0/1	0/1	0	0	1
MpDMATvfSdpMasterMem5Error	0x29	0/1	0/1	0/1	0	0	1
MpDMATvfSdpMasterMem6Error	0x2a	0/1	0/1	0/1	0	0	1
sdp_watchdog_timeout_error	0x2b	1	1	1	0	0	1
MpDMApteCmdfifoMemError	0x2c	0/1	0/1	0/1	0	0	1

MpDMA Pte HubDataFifo MemError	0x2d	0/1	0/1	0/1	0	0	1
MpDMA Pte InternalDataFifo MemError	0x2e	0/1	0/1	0/1	0	0	1
MpDMA Pte CmdmemDMAFifo MemError	0x2f	0/1	0/1	0/1	0	0	1
MpDMA Pte CmdmemInternalFifo MemError	0x30	0/1	0/1	0/1	0	0	1
MpDMA Tvf SdpMasterMem7Error	0x31	0/1	0/1	0/1	0	0	1

MSR0000_0452...MSRC000_2142 [MPDMA Machine Check Address] (MCA::MPDMA::MCA_ADDR_MPDMA)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

MCA::MPDMA::MCA_ADDR_MPDMA stores an address and other information associated with the error in MCA::MPDMA::MCA_STATUS_MPDMA. The register is only meaningful if MCA::MPDMA::MCA_STATUS_MPDMA[Val]=1 and MCA::MPDMA::MCA_STATUS_MPDMA[AddrV]=1.

_instMPDMAMPDMA1_n[5:0]_aliasMSRLEGACY; MSR0000_0452

_instMPDMAMPDMA1_n[5:0]_aliasMSR; MSRC000_2142

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::MPDMA::MCA_STATUS_MPDMA. For physical addresses, the most significant bit is given by Core::X86::CpuId::LongModeInfo[PhysAddrSize].

Table 122: MCA_ADDR_MPDMA

Error Type	Bits	Description
MpDMASram0Error	[55:0]	Reserved
MpDMASram1Error	[55:0]	Reserved
MpDMASram2Error	[55:0]	Reserved
MpDMASram3Error	[55:0]	Reserved
MpDMADCacheAError	[55:0]	Reserved
MpDMADCacheBError	[55:0]	Reserved
MpDMADTagAError	[55:0]	Reserved
MpDMADTagBError	[55:0]	Reserved
MpDMAICacheAError	[55:0]	Reserved
MpDMAICacheBError	[55:0]	Reserved
MpDMAITagAError	[55:0]	Reserved
MpDMAITagBError	[55:0]	Reserved
MpDMADCacheAError_1	[55:0]	Reserved
MpDMADCacheBError_1	[55:0]	Reserved
MpDMADTagAError_1	[55:0]	Reserved
MpDMADTagBError_1	[55:0]	Reserved
MpDMAICacheAError_1	[55:0]	Reserved

MpDMAICacheBError_1	[55:0]	Reserved
MpDMAITagAError_1	[55:0]	Reserved
MpDMAITagBError_1	[55:0]	Reserved
MpDMADCacheAError_2	[55:0]	Reserved
MpDMADCacheBError_2	[55:0]	Reserved
MpDMADTagAError_2	[55:0]	Reserved
MpDMADTagBError_2	[55:0]	Reserved
MpDMAICacheAError_2	[55:0]	Reserved
MpDMAICacheBError_2	[55:0]	Reserved
MpDMAITagAError_2	[55:0]	Reserved
MpDMAITagBError_2	[55:0]	Reserved
MpDMASHubIfRdBufError	[55:0]	Reserved
MpDMATvfDVSecError	[55:0]	Reserved
MpDMATvfMMioMbox0Error	[55:0]	Reserved
MpDMATvfMmioMbox1Error	[55:0]	Reserved
MpDMATvfDoorbellMemError	[55:0]	Reserved
MpDMATvfSdpSlaveMem0Error	[55:0]	Reserved
MpDMATvfSdpSlaveMem1Error	[55:0]	Reserved
MpDMATvfSdpSlaveMem2Error	[55:0]	Reserved
MpDMATvfSdpMasterMem0Error	[55:0]	Reserved
MpDMATvfSdpMasterMem1Error	[55:0]	Reserved
MpDMATvfSdpMasterMem2Error	[55:0]	Reserved
MpDMATvfSdpMasterMem3Error	[55:0]	Reserved
MpDMATvfSdpMasterMem4Error	[55:0]	Reserved
MpDMATvfSdpMasterMem5Error	[55:0]	Reserved
MpDMATvfSdpMasterMem6Error	[55:0]	Reserved
sdp_watchdog_timeout_error	[55:0]	Reserved
MpDMA PteCmdfifoMemError	[55:0]	Reserved
MpDMA PteHubDataFifoMemError	[55:0]	Reserved
MpDMA PteInternalDataFifoMemError	[55:0]	Reserved
MpDMA PteCmdmemDMAFifoMemError	[55:0]	Reserved
MpDMA PteCmdmemInternalFifoMemError	[55:0]	Reserved
MpDMATvfSdpMasterMem7Error	[55:0]	Reserved

**MSR0000_0453...MSRC000_2143 [MPDMA Machine Check Miscellaneous 0]
(MCA::MPDMA::MCA_MISC0_MPDMA)**

Log miscellaneous information associated with errors.

_instMPDMAMPDMA1_n[5:0]_aliasMSRLEGACY; MSR0000_0453

_instMPDMAMPDMA1_n[5:0]_aliasMSR; MSRC000_2143

Bits	Description
63	Valid. Reset: 1. 1=A valid CntP field is present in this register. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
62	CntP. Reset: 1. 1=A valid threshold counter is present. AccessType: Core::X86::Msr::HWCR[McStatusWrEn] ? Read-write : Read-only.
61	Locked. 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	IntP. 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::MPDMA::MCA_MISC0_MPDMA[Locked]) ? Read-write : Read-only.
59:56	Reserved.
55:52	LvtOffset. 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::MPDMA::MCA_MISC0_MPDMA[Locked]) ? Read-write : Read-only.
51	CntEn. Reset: 0. 1=Count thresholding errors. AccessType: (Core::X86::Msr::HWCR[McStatusWrEn] !MCA::MPDMA::MCA_MISC0_MPDMA[Locked]) ? Read-write : Read-only.
50:49	ThresholdIntType. 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::MPDMA::MCA_MISC0_MPDMA[Locked]) ? Read-write : Read-only.
48	Ovrflw. 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::MPDMA::MCA_MISC0_MPDMA[Locked]) ? Read-write : Read-only.
47:44	Reserved.
43:32	ErrCnt. 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::MPDMA::MCA_MISC0_MPDMA[Locked]) ? Read-write : Read-only.
31:24	BlkPtr. Read-write. Reset: 00h. 00h=Extended MISC MSR block is not valid. 01h=Extended MSR block is valid.
23:0	Reserved.

MSRC000_2144 [MPDMA Machine Check Configuration] (MCA::MPDMA::MCA_CONFIG_MPDMA)

Reset: 0000_0000_0000_0121h.

Controls configuration of the associated machine check bank.

_instMPDMAMPDMA1_n[5:0]_aliasMSR; MSRC000_2144

Bits	Description
63:41	Reserved.
40	IntEn. Read-write. Reset: 0. Init: BIOS,0. 1=When set, this bank will generate corrected error interrupts.
39	Reserved.
38:37	DeferredIntType. 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	McaXEnable. 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	IntPresent. Read-only, Volatile . Reset: 0. 1=This bank can be configured to trigger a corrected error interrupt using MCA::MPDMA::MCA_CONFIG_MPDMA[IntEn].
9	McaFruTextInMca. Read-write. Reset: 0. Init: BIOS,0. 1=FruText is reported McaSynd1/McaSynd2 registers
8	McaLsbInStatusSupported. Read-only. Reset: 1. 1=MCA::MPDMA::MCA_CONFIG_MPDMA[McaLsbInStatusSupported] indicates that AddrLsb is located in McaStatus registers.
7:6	Reserved.
5	DeferredIntTypeSupported. Read-only. Reset: 1. 1=MCA::MPDMA::MCA_CONFIG_MPDMA[DeferredIntType] controls the type of interrupt generated on a deferred error. Deferred errors are supported in this bank only if MCA::MPDMA::MCA_CONFIG_MPDMA[DeferredErrorLoggingSupported]=1.
4:3	Reserved.
2	DeferredErrorLoggingSupported. 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	McaX. Read-only. Reset: 1. 1=This bank provides Machine Check Architecture Extensions. Up to 4 additional MISC registers (MISC1-MISC4) are supported. MCA::MPDMA::MCA_MISC0_MPDMA[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::MPDMA::MCA_STATUS_MPDMA[TCC] is present.

MSRC000_2145 [MPDMA IP Identification] (MCA::MPDMA::MCA_IPID_MPDMA)

Reset: 0003_0001_0000_0000h.

The MCA::MPDMA::MCA_IPID_MPDMA register is used by software to determine what IP type and revision is associated with the MCA bank.

_instMPDMAMPDMA1_n[5:0]_aliasMSR; MSRC000_2145

Bits	Description
63:48	McaType . Read-only. Reset: 0003h. The McaType of the MCA bank within this IP.
47:44	InstanceIdHi . 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	HardwareID . Read-only. Reset: 001h. The Hardware ID of the IP associated with this MCA bank.
31:0	InstanceId . Read-write. Reset: 0000_0000h. The instance ID of this IP. This is initialized to a unique ID per instance of this register.
	Init: _instMPDMAMPDMA1_n0_aliasMSR: 0CF3_0400h
	Init: _instMPDMAMPDMA1_n1_aliasMSR: 0E63_0400h
	Init: _instMPDMAMPDMA1_n2_aliasMSR: 0ED3_0400h
	Init: _instMPDMAMPDMA1_n3_aliasMSR: 0F43_0400h
	Init: _instMPDMAMPDMA1_n4_aliasMSR: 0953_0400h
	Init: _instMPDMAMPDMA1_n5_aliasMSR: 0983_0400h

MSRC000_2146 [MPDMA Machine Check Syndrome] (MCA::MPDMA::MCA_SYND_MPDMA)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::MPDMA::MCA_STATUS_MPDMA [Thread 0](#)

_instMPDMAMPDMA1_n[5:0]_aliasMSR; MSRC000_2146

Bits	Description
63:27	Reserved.
26:24	ErrorPriority . Read-write, Volatile . Reset: Cold, 0h. Encodes the priority of the error logged in MCA::MPDMA::MCA_SYND_MPDMA. 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	Length . 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	ErrorInformation . Read-write, Volatile . Reset: Cold, 0_0000h. Contains error-specific information about the location of the error. Decoding is available in Table 123 [MCA_SYND_MPDMA].

Table 123: MCA_SYND_MPDMA

Error Type	Bits	Description
MpDMASram0Error	[17:8] [7:0]	Reserved Reserved
MpDMASram1Error	[17:8] [7:0]	Reserved Reserved
MpDMASram2Error	[17:8] [7:0]	Reserved Reserved
MpDMASram3Error	[17:8] [7:0]	Reserved Reserved
MpDMADCacheAError	[17:8] [7:0]	Reserved Reserved
MpDMADCacheBError	[17:8] [7:0]	Reserved Reserved
MpDMADTagAError	[17:7] [6:0]	Reserved Reserved
MpDMADTagBError	[17:7] [6:0]	Reserved Reserved

MpDMAICacheAError	[17:8] [7:0]	Reserved Reserved
MpDMAICacheBError	[17:8] [7:0]	Reserved Reserved
MpDMAITagAError	[17:6] [5:0]	Reserved Reserved
MpDMAITagBError	[17:6] [5:0]	Reserved Reserved
MpDMADCacheAError_1	[17:8] [7:0]	Reserved Reserved
MpDMADCacheBError_1	[17:8] [7:0]	Reserved Reserved
MpDMADTagAError_1	[17:7] [6:0]	Reserved Reserved
MpDMADTagBError_1	[17:7] [6:0]	Reserved Reserved
MpDMAICacheAError_1	[17:8] [7:0]	Reserved Reserved
MpDMAICacheBError_1	[17:8] [7:0]	Reserved Reserved
MpDMAITagAError_1	[17:6] [5:0]	Reserved Reserved
MpDMAITagBError_1	[17:6] [5:0]	Reserved Reserved
MpDMADCacheAError_2	[17:8] [7:0]	Reserved Reserved
MpDMADCacheBError_2	[17:8] [7:0]	Reserved Reserved
MpDMADTagAError_2	[17:7] [6:0]	Reserved Reserved
MpDMADTagBError_2	[17:7] [6:0]	Reserved Reserved
MpDMAICacheAError_2	[17:8] [7:0]	Reserved Reserved
MpDMAICacheBError_2	[17:8] [7:0]	Reserved Reserved
MpDMAITagAError_2	[17:6] [5:0]	Reserved Reserved
MpDMAITagBError_2	[17:6] [5:0]	Reserved Reserved
MpDMASHubIfRdBufError	[17:6] [5:0]	Reserved Reserved
MpDMATvfDVSecError	[17:8] [7:0]	Reserved Reserved
MpDMATvfMMioMbox0Error	[17:9] [8:0]	Reserved Reserved
MpDMATvfMmioMbox1Error	[17:9] [8:0]	Reserved Reserved
MpDMATvfDoorbellMemError	[17:9]	Reserved

	[8:0]	Reserved
MpDMATvfSdpSlaveMem0Error	[17:4] [3:0]	Reserved Reserved
MpDMATvfSdpSlaveMem1Error	[17:4] [3:0]	Reserved Reserved
MpDMATvfSdpSlaveMem2Error	[17:5] [4:0]	Reserved Reserved
MpDMATvfSdpMasterMem0Error	[17:7] [6:0]	Reserved Reserved
MpDMATvfSdpMasterMem1Error	[17:6] [5:0]	Reserved Reserved
MpDMATvfSdpMasterMem2Error	[17:5] [4:0]	Reserved Reserved
MpDMATvfSdpMasterMem3Error	[17:4] [3:0]	Reserved Reserved
MpDMATvfSdpMasterMem4Error	[17:6] [5:0]	Reserved Reserved
MpDMATvfSdpMasterMem5Error	[17:4] [3:0]	Reserved Reserved
MpDMATvfSdpMasterMem6Error	[17:6] [5:0]	Reserved Reserved
sdp_watchdog_timeout_error	[17:7] [6:0]	Reserved Reserved
MpDMA PteCmdfifoMemError	[17:7] [6:0]	Reserved Reserved
MpDMA PteHubDataFifoMemError	[17:10] [9:0]	Reserved Reserved
MpDMA PteInternalDataFifoMemError	[17:9] [8:0]	Reserved Reserved
MpDMA PteCmdmemDMAFifoMemError	[17:5] [4:0]	Reserved Reserved
MpDMA PteCmdmemInternalFifoMemError	[17:7] [6:0]	Reserved Reserved
MpDMATvfSdpMasterMem7Error	[17:5] [4:0]	Reserved Reserved

MSRC001_0414 [MPDMA Machine Check Control Mask] (MCA::MPDMA::MCA_CTL_MASK_MPDMA)

Read-write. Reset: 0000_0000_0000_0000h.

Inhibit detection of an error source.

_instMPDMAMPDMA1_n[5:0]_aliasMSR; MSRC001_0414

Bits	Description
63:49	Reserved.
48	MpDMAPteCmdmemInternalFifoMemError . Read-write. Reset: 0. MPDMA PTE Command Memory Internal ECC or parity error
47	MpDMAPteCmdmemDMAFifoMemError . Read-write. Reset: 0. MPDMA PTE Command Memory DMA ECC or parity error
46	MpDMAPteInternalDataFifoMemError . Read-write. Reset: 0. MPDMA PTE Internal Data FIFO ECC or parity error
45	MpDMAPteHubDataFifoMemError . Read-write. Reset: 0. MPDMA PTE Hub Data FIFO ECC or parity error
44	MpDMAPteCmdfifoMemError . Read-write. Reset: 0. MPDMA PTE Command FIFO ECC or parity error
43	sdp_watchdog_timeout_error . Read-write. Reset: 0. sdp watchdog timer expired
42:29	Reserved.
28	MpDMASHubIfRdBufError . Read-write. Reset: 0. System Hub Read Buffer ECC or parity error.
27	MpDMAITagBError_2 . Read-write. Reset: 0. Instruction Tag Cache Bank B ECC or parity error.
26	MpDMAITagAError_2 . Read-write. Reset: 0. Instruction Tag Cache Bank A ECC or parity error.
25	MpDMAICacheBError_2 . Read-write. Reset: 0. Instruction Cache Bank B ECC or parity error.
24	MpDMAICacheAError_2 . Read-write. Reset: 0. Instruction Cache Bank A ECC or parity error.
23	MpDMADTagBError_2 . Read-write. Reset: 0. Data Tag Cache Bank B ECC or parity error.
22	MpDMADTagAError_2 . Read-write. Reset: 0. Data Tag Cache Bank A ECC or parity error.
21	MpDMADCacheBError_2 . Read-write. Reset: 0. Data Cache Bank B ECC or parity error.
20	MpDMADCacheAError_2 . Read-write. Reset: 0. Data Cache Bank A ECC or parity error.
19	MpDMAITagBError_1 . Read-write. Reset: 0. Instruction Tag Cache Bank B ECC or parity error.
18	MpDMAITagAError_1 . Read-write. Reset: 0. Instruction Tag Cache Bank A ECC or parity error.
17	MpDMAICacheBError_1 . Read-write. Reset: 0. Instruction Cache Bank B ECC or parity error.
16	MpDMAICacheAError_1 . Read-write. Reset: 0. Instruction Cache Bank A ECC or parity error.
15	MpDMADTagBError_1 . Read-write. Reset: 0. Data Tag Cache Bank B ECC or parity error.
14	MpDMADTagAError_1 . Read-write. Reset: 0. Data Tag Cache Bank A ECC or parity error.
13	MpDMADCacheBError_1 . Read-write. Reset: 0. Data Cache Bank B ECC or parity error.
12	MpDMADCacheAError_1 . Read-write. Reset: 0. Data Cache Bank A ECC or parity error.
11	MpDMAITagBError . Read-write. Reset: 0. Instruction Tag Cache Bank B ECC or parity error.
10	MpDMAITagAError . Read-write. Reset: 0. Instruction Tag Cache Bank A ECC or parity error.
9	MpDMAICacheBError . Read-write. Reset: 0. Instruction Cache Bank B ECC or parity error.
8	MpDMAICacheAError . Read-write. Reset: 0. Instruction Cache Bank A ECC or parity error.
7	MpDMADTagBError . Read-write. Reset: 0. Data Tag Cache Bank B ECC or parity error.
6	MpDMADTagAError . Read-write. Reset: 0. Data Tag Cache Bank A ECC or parity error.
5	MpDMADCacheBError . Read-write. Reset: 0. Data Cache Bank B ECC or parity error.
4	MpDMADCacheAError . Read-write. Reset: 0. Data Cache Bank A ECC or parity error.
3	MpDMASram3Error . Read-write. Reset: 0. Main SRAM [127:96] bank ECC or parity error.
2	MpDMASram2Error . Read-write. Reset: 0. Main SRAM [95:64] bank ECC or parity error.
1	MpDMASram1Error . Read-write. Reset: 0. Main SRAM [63:32] bank ECC or parity error.
0	MpDMASram0Error . Read-write. Reset: 0. Main SRAM [31:0] bank ECC or parity error.

MSRC000_214E [MPDMA Machine Check Syndrome Extended] (MCA::MPDMA::MCA_SYND1_MPDMA)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::MPDMA::MCA_STATUS_MPDMA [Thread 0](#)

_instMPDMAMPDMA1_n[5:0]_aliasMSR; MSRC000_214E

Bits	Description
63:0	Syndrone. Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::MPDMA::MCA_SYND1_MPDMA register stores information associated with the error in MCA::MPDMA::MCA_STATUS_MPDMA or MCA_DESTAT. The register is meaningful if MCA::MPDMA::MCA_STATUS_MPDMA[SyndV]=1. When MCA::MPDMA::MCA_CONFIG_MPDMA[McaFruTextInMca]=1, MCA::MPDMA::MCA_SYND1_MPDMA stores ASCII FruText associated with the error.

MSRC000_214F [MPDMA Machine Check Syndrome Extended] (MCA::MPDMA::MCA_SYND2_MPDMA)

Read-write, Volatile. Reset: Cold, 0000_0000_0000_0000h.

Logs physical location information associated with error in MCA::MPDMA::MCA_STATUS_MPDMA [Thread 0](#)

_instMPDMAMPDMA1_n[5:0]_aliasMSR; MSRC000_214F

Bits	Description
63:0	Syndrone. Read-write, Volatile . Reset: Cold, 0000_0000_0000_0000h. The MCA::MPDMA::MCA_SYND2_MPDMA register stores information associated with the error in MCA::MPDMA::MCA_STATUS_MPDMA or MCA_DESTAT. The register is meaningful if MCA::MPDMA::MCA_STATUS_MPDMA[SyndV]=1. When MCA::MPDMA::MCA_CONFIG_MPDMA[McaFruTextInMca]=1, MCA::MPDMA::MCA_SYND2_MPDMA stores ASCII FruText associated with the error.

3.3 PCI Express® RAS Features

3.3.1 PCI Express® Advanced Error Reporting (AER)

Each of the [PCIe](#) root ports support AER functionality. This applies to both the physical root ports used to connect to external devices, as well as the internal root ports used to connect internal controllers.

The physical root ports do not support the following optional AER features:

- Receiver Buffer Overflow
- Flow Control Protocol Error
- Surprise Down

The internal root ports do not support the following optional AER features:

- Poisoned TLP Egress Blocking
- Receiver Buffer Overflow
- Flow Control Protocol Error
- Surprise Down

For poisoned TLP error injection:

- RX:
1. Write 1 to PCIE_ERR_CNTL.RX_GENERATE_POIS_TLP[6]
 2. This will simulate receiving a TLP with the poisoned bit (EP bit) set to 1 on the next incoming TLP that has data-payload.

TX:

1. Write 1 to PCIE_TX_ERR_CTRL.TX_GENERATE_POIS_TLP[2]

2. This will force the TX to set the the poisoned bit (EP bit) 1 on the next out-bound TLP that has data-payload.

3.3.1.1 Recommended PCI Express® AER Severity

It is recommended that SBIOS set the initial [PCIe](#) AER severity throughout the PCIe topology as per the following tables. These may be subsequently modified by the operating system. The processor root ports and internal endpoints do not support reporting all of the error types listed below, but those errors may still be reported by platform level switches and endpoints.

Table 124: AER Severity for Root Complex and Switch Downstream Ports

AER Feature	Recommended Severity	Mask
Data Link Protocol Error	Fatal	0
Surprise Down Error	Fatal	0
Poisoned TLP Received	Advisory Non-Fatal	0
Flow Control Protocol Error	Fatal	0
Completion Timeout Error	Fatal	0
Completer Abort	Advisory Non-Fatal	0
Unexpected Completion Error	Fatal	0
Receiver Overflow Status	Fatal	0
Malformed TLP	Fatal	0
ECRC Error	Fatal	0
Unsupported Request Error	Advisory Non-Fatal	0
ACS Violation	Fatal	0
Uncorrectable Internal Error	Fatal	0
MC Blocked TLP	Advisory Non-Fatal	0
AtomicOp Egress Blocked	Fatal	0
TLP Prefix Blocked Error	Fatal	0
Poisoned TLP Egress Blocked	Fatal	0
Receiver Error	Correctable	0
Bad TLP	Correctable	0
Bad DLLP	Correctable	0
REPLAY_NUM Rollover	Correctable	0
Replay Timer Timeout	Correctable	0
PCRC Check Failed	Advisory Non-Fatal	0
Misrouted IDE TLP	Advisory Non-Fatal	0
IDE Check Failed	Fatal	0
Advisory Non-Fatal Error	Correctable	1
Corrected Internal Error	Correctable	0
Header Log Overflow	Correctable	0

Table 125: AER Severity for Endpoint Devices and Switch Upstream Ports

AER Feature	Recommended Severity	Mask
Data Link Protocol Error	Fatal	0
Surprise Down Error	Fatal	0
Poisoned TLP Received	Advisory Non-Fatal	0

Flow Control Protocol Error	Fatal	0
Completion Timeout Error	Fatal	0
Completer Abort	Advisory Non-Fatal	0
Unexpected Completion Error	Fatal	0
Receiver Overflow Status	Fatal	0
Malformed TLP	Fatal	0
ECRC Error	Fatal	0
PCRC Check Failed	Advisory Non-Fatal	0
Misrouted IDE TLP	Advisory Non-Fatal	0
IDE Check Failed	Fatal	0
Unsupported Request Error	Advisory Non-Fatal	0
ACS Violation	Fatal	0
MC Blocked TLP	Fatal	0
Uncorrectable Internal Error	Fatal	0
AtomicOp Egress Blocked	Fatal	0
TLP Prefix Blocked Error	Fatal	0
Poisoned TLP Egress Blocked	Fatal	0
Receiver Error	Correctable	0
Bad TLP	Correctable	0
Bad DLLP	Correctable	0
REPLAY_NUM Rollover	Correctable	0
Replay Timer Timeout	Correctable	0
Advisory Non-Fatal Error	Correctable	1
Corrected Internal Error	Correctable	0
Header Log Overflow	Correctable	0

3.3.2 PCIe® Error Status

[PCIe](#) errors are logged in standard AER status registers.

3.3.3 PCI Express® eDPC

eDPC is supported on the externally facing [PCIe](#) root ports used to connect external PCIe devices.

3.3.4 PCI Express® ECRC

The processor supports PCI Express® ECRC generation and checking on all [PCIe](#) root ports. ECRC is not preserved for peer-to-peer requests that cross through the processor. In general, ECRC generation capabilities should only be enabled in root ports and endpoints if there is a switch between them.

Note that it is legal for the processor to send requests with ECRC to endpoints that do not support ECRC. This may occur if a root port connects to a switch with multiple downstream ports where some ports connect to endpoint devices supporting ECRC and others connect to endpoints that do not support ECRC. Endpoints that do not support ECRC should ignore the ECRC information.

The processor implements a standard PCI Express ECRC control as part of AER located at offset 0x168 within each PCI Express bridge. Please refer to the PCI Express Base Specification section 7.10.7 for more details on the registers used to

control ECRC.

The following algorithm may be used to determine when and where to enable ECRC. System bios should enable ECRC before handing off control to the OS.

For each PCIE root port, check the device immediately behind the bridge.

If this is an endpoint device, do not enable ECRC on the root port and go to the next root port

Else if this is a PCI Express bridge and there are additional PCI Express bridges located on the bus behind it, the root port is connected to a PCI Express switch (as opposed to a different type of bridge such as a PCI Express to PCI-X bridge).

Search all PCI Express endpoint devices below the switch (which may be below additional switches).

If any endpoint device contains an AER capability in the extended capability list (type 0x1) and sets the ECRC Check Capable bit in Advanced Error Capabilities and Control Register (offset 0x18 bit[8] inside the device's AER capability) and the root port has ECRC_GEN_CAP = 0x1.

Set ECRC Check Enable in the endpoint device (offset 0x18 bit[7] inside the device's AER capability).

Set ECRC Generation Enable in the root port (root port offset 0x168 bit[6]).

If any endpoint device contains an AER capability in the extended capability list (type 0x1) and sets the ECRC Generation Capable bit in Advanced Error Capabilities and Control Register (offset 0x18 bit[5] inside the AER cap) and the root port has ECRC_CHECK_CAP = 0x1.

Set ECRC Check Enable in the root port (root port offset 0x168 bit[8]).

Set ECRC Generation Enable in the endpoint device (offset 0x18 bit[6] inside the device's AER capability).