# System Event Log (SEL)

## *Troubleshooting Guide*

Summary and Definition of Events Generated by Intel® Server Board D50TNP and M50CYP Families.

**Rev 1.0**

**June 2021**

<Blank page>

# *Document Revision History*

| Date | Revision | Changes |
|---|---|---|
| **June 2021** | 1.0 | Initial release. |

# *Disclaimers*

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at intel.com.

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

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

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

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

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

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

Intel, Xeon and the Intel logo are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

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

© Intel Corporation

# Table of Contents

# List of Tables

# 1. Introduction

The server management hardware that is part of the Intel® Server Boards and Intel® Server Platforms serves as a vital part of the overall server management strategy. The server management hardware provides essential information to the system administrator and provides the administrator the ability to remotely control the server, even when the operating system is not running.

The Intel® Server Boards and Intel® Server Platforms offer comprehensive hardware and software based solutions. The server management features make the servers simple to manage and provide alerting on system events. From entry to enterprise systems, good overall server management is essential to reduce overall total cost of ownership.

This troubleshooting guide is intended to help the users better understand the events that are logged in the Baseboard Management Controllers (BMC) System Event Logs (SEL) on these Intel® Server Boards.

There is a separate user guide that covers the general server management and the server management software offered on the Intel® Server Boards and Intel® Server Platforms.

This document supports the following Intel® Server products currently supported by this document:

- Intel® Server System D50TNP and M50CYP families supporting the 3rd Gen Intel® Xeon® Scalable processor family.

## 1.1 Purpose

The purpose of this document is to list all possible events generated by the Intel platform. It may be possible that other sources (not under Intel's control) also generate events, which are not described in this document.

## 1.2 Industry Standard

### 1.2.1 Intelligent Platform Management Interface (IPMI)

The key characteristic of the Intelligent Platform Management Interface (IPMI) is that the inventory, monitoring, logging, and recovery control functions are available independently of the main processors, BIOS, and operating system. Platform management functions can also be made available when the system is in a power-down state.

IPMI works by interfacing with the BMC, which extends management capabilities in the server system and operates independently of the main processor by monitoring the on-board instrumentation. Through the BMC, IPMI also allows administrators to control power to the server and remotely access BIOS configuration and operating system console information.

IPMI defines a common platform instrumentation interface to enable interoperability between

- The baseboard management controller and chassis;
- The baseboard management controller and systems management software; and
- Between servers.

IPMI enables the following:

- Common access to platform management information, consisting of
    - Local access from systems management software;
    - Remote access from LAN;
    - Inter-chassis access from Intelligent Chassis Management Bus; and
    - Access from LAN, serial/modem, IPMB, PCI SMBus*, or ICMB, available even if the processor is down.

- Isolation of systems management software from hardware.
- Ability to make hardware advancements without impacting the systems management software.
- Facilitation of cross-platform management software.

Find more information on IPMI at http://www.intel.com/design/servers/ipmi.

### 1.2.2 Baseboard Management Controller (BMC)

A baseboard management controller (BMC) is a specialized microcontroller embedded on most Intel Server Boards. The BMC is the heart of the IPMI architecture and provides the intelligence behind intelligent platform management, that is, the autonomous monitoring and recovery features implemented directly in platform management hardware and firmware.

Different types of sensors built into the computer system report to the BMC on parameters such as temperature, cooling fan speeds, power mode, operating system status, and so on. The BMC monitors the system for critical events by communicating with various sensors on the system board; it sends alerts and logs events when certain parameters exceed their preset thresholds, indicating a potential failure of the system. The administrator can also remotely communicate with the BMC to take some corrective action such as resetting or power cycling the system to get a hung OS running again. These abilities save on the total cost of ownership of a system.

For Intel Server Boards and Intel Server Platforms, the BMC supports the industry standard *IPMI 2.0 Specification*, enabling remote configuration, monitoring, and systems recovery.

#### 1.2.2.1 System Event Log (SEL)

The BMC provides a centralized, non-volatile repository for critical, warning, and informational system events called the System Event Log (SEL). By having the BMC manage the SEL and logging functions, it helps to ensure that “post-mortem” logging information is available if a failure occurs that disables the system processor(s).

The BMC allows access to the SEL from in-band and out-of-band mechanisms. There are various tools and utilities that can be used to access the SEL including the SELView utility and multiple open sourced IPMI tools.

### 1.2.3 Intel® Intelligent Power Node Manager Version 3.0

Intel® Intelligent Power Node Manager version 3.0 is a platform-resident technology that enforces power and thermal policies for the platform. These policies are applied by exploiting subsystem knobs (such as processor P and T states) that can be used to control power consumption. Intel® Intelligent Power Node Manager enables data center power and thermal management by exposing an external interface to management software through which platform policies can be specified. It also enables specific data center power management usage models such as power limiting.

The configuration and control commands are used by the external management software or BMC to configure and control the Intel® Intelligent Power Node Manager feature. Because the platform firmware does not have any external interface, external commands are first received by the BMC over LAN and then relayed to the platform firmware over Intelligent Power Management Bus (IPMB) channel. The BMC acts as a relay and the transport conversion device for these commands. For simplicity, the commands from the management console might be encapsulated in a generic CONFIG packet format (configuration data length, configuration data blob) to the BMC so that the BMC does not even have to parse the actual configuration data.

The BMC provides the access point for remote commands from external management software and generates alerts to them. Intel® Intelligent Power Node Manager on Intel® Management Engine (Intel® ME) is

an IPMI satellite controller. A mechanism exists to forward commands to Intel® ME and then sends the response back to originator. Similarly events from Intel® ME are sent as alerts outside of the BMC.

# 2. Basic Decoding of a SEL Record

The System Event Log (SEL) record format is defined in the *IPMI Specification*. The following section provides a basic definition for each of the fields in a SEL. For more details, see the *IPMI Specification*.

The definitions for the standard SEL can be found in Table 1.

The definitions for the OEM defined event logs can be found in Table 3 and Table 4.

## 2.1 Default Values in the SEL Records

Unless otherwise noted in the event record description, the following are the default values for all SEL entries.

- Byte [3] = Record Type (RT) = 02h = System event record
- Byte [9:8] = Generator ID = 0020h = BMC firmware
- Byte [10] = Event Message Revision (ER) = 04h = IPMI 2.0

**Table 1. SEL Record Format**

| Byte | Field | Description |
|---|---|---|
| **1, 2** | Record ID (RID) | ID used for SEL record access. |
| **3** | Record Type (RT) | [7:0] – Record type<br>02h = System event record (default)<br>C0h-DFh = OEM timestamped, bytes 8–16 OEM defined (see Table 3)<br>E0h-FFh = OEM non-timestamped, bytes 4–16 OEM defined (see Table 4) |
| **4–7** | Timestamp (TS) | Time when the event was logged. The least significant byte is first.<br>For example, TS:[29][76][68][4C] = 4C687629h = 1281914409 = Sun, 15 Aug 2010 23:20:09 UTC<br>**Note**: There are various websites that convert the raw number to a date/time. |
| **8, 9** | Generator ID (GID) | RqSA and LUN if event was generated from IPMB.<br>Software ID if event was generated from system software.<br><br>*Byte 1*<br>[7:1] – 7-bit I2C slave address, or 7-bit system software ID<br>[0] – 0b = ID is IPMB slave address, 1b = System software ID<br>Software ID values:<br>0001h – BIOS POST for POST errors, RAS configuration/state, timestamp synch, OS boot events<br>0033h – BIOS SMI handler<br>0020h – BMC firmware (default)<br>002Ch – Intel® ME firmware<br>0041h – Server management software<br>00C0h – HSC firmware – HSBP A<br>00C2h – HSC firmware – HSBP B<br><br>*Byte 2*<br>[7:4] – Channel number. Channel that event message was received over. 0h if the event message was received from the system interface, primary IPMB, or internally generated by the BMC.<br>[3:2] – Reserved. Write as 00b.<br>[1:0] – IPMB device LUN if byte 1 holds slave address. 00b otherwise. |
| **10** | EvM Rev (ER) | Event message format version.<br>04h = IPMI v2.0 (default)<br>03h = IPMI v1.0 |
| **11** | Sensor Type (ST) | Sensor type code for sensor that generated the event. |
| **12** | Sensor # (SN) | Number of sensors that generated the event (from SDR). |

| Byte | Field | Description |
|---|---|---|
| **13** | Event Dir/Event Type (EDIR) | *Event Dir*<br>[7] – 0b = Assertion event, 1b = Deassertion event.<br><br>*Event Type*<br>Type of trigger for the event; for example, critical threshold going high, state asserted, and so on. Also indicates class of the event; for example, discrete, threshold, or OEM. The Event Type field is encoded using the Event/Reading Type Code.<br>[6:0] – Event Type Codes<br>01h = Threshold (states = 0x00-0x0b)<br>02h-0ch = Discrete<br>6Fh = Sensor-specific<br>70-7Fh = OEM |
| **14** | Event Data 1 (ED1) | See Table 2. |
| **15** | Event Data 2 (ED2) | |
| **16** | Event Data 3 (ED3) | |

## Table 2. Event Request Message Event Data Field Contents

| Sensor Class | Event Data |
|---|---|
| **Threshold** | *Event Data 1*<br>[7:6] –<br>00b = Unspecified Event Data 2<br>01b = Trigger reading in Event Data 2<br>10b = OEM code in Event Data 2<br>11b = Sensor-specific event extension code in Event Data 2<br>[5:4] –<br>00b = Unspecified Event Data 3<br>01b = Trigger threshold value in Event Data 3<br>10b = OEM code in Event Data 3<br>11b = Sensor-specific event extension code in Event Data 3<br>[3:0] – Offset from Event/Reading Code for threshold event.<br><br>*Event Data 2* – Reading that triggered event, FFh or not present if unspecified.<br>*Event Data 3* – Threshold value that triggered event, FFh or not present if unspecified. If present, Event Data 2 must be present. |
| **Discrete** | *Event Data 1*<br>[7:6] –<br>00b = Unspecified Event Data 2<br>01b = Previous state and/or severity in Event Data 2<br>10b = OEM code in Event Data 2<br>11b = Sensor-specific event extension code in Event Data 2<br>[5:4] –<br>00b = Unspecified Event Data 3<br>01b = Reserved<br>10b = OEM code in Event Data 3<br>11b = Sensor-specific event extension code in Event Data 3<br>[3:0] – Offset from Event/Reading Code for discrete event state<br><br>*Event Data 2*<br>[7:4] – Optional offset from "Severity" Event/Reading Code (0Fh if unspecified).<br>[3:0] – Optional offset from Event/Reading Type Code for previous discrete event state (0Fh if unspecified).<br><br>*Event Data 3* – Optional OEM code. FFh or not present if unspecified. |

| Sensor Class | Event Data |
|---|---|
| **OEM** | *Event Data 1*<br>[7:6] –<br>00b = Unspecified in Event Data 2<br>01b = Previous state and/or severity in Event Data 2<br>10b = OEM code in Event Data 2<br>11b = Reserved<br>[5:4] –<br>00b = Unspecified Event Data 3<br>01b = Reserved<br>10b = OEM code in Event Data 3<br>11b = Reserved<br>[3:0] – Offset from Event/Reading Type Code<br><br>*Event Data 2*<br>[7:4] – Optional OEM code bits or offset from "Severity" Event/Reading Type Code (0Fh if unspecified).<br>[3:0] – Optional OEM code or offset from Event/Reading Type Code for previous event state (0Fh if unspecified).<br><br>*Event Data 3* – Optional OEM code. FFh or not present if unspecified. |

## Table 3. OEM SEL Record (Type C0h-Dfh)

| Byte | Field | Description |
|---|---|---|
| **1, 2** | Record ID (RID) | ID used for SEL Record access. |
| **3** | Record Type (RT) | [7:0] – Record Type<br>C0h-DFh = OEM timestamped, bytes 8–16 OEM defined |
| **4–7** | Timestamp (TS) | Time when the event was logged. The least significant byte is first.<br>For example, TS:[29][76][68][4C] = 4C687629h = 1281914409 = Sun, 15 Aug 2010 23:20:09 UTC<br>**Note**: There are various websites that convert the raw number to a date/time. |
| **8–10** | Manufacturer ID | The least significant byte is first. The manufacturer ID is a 20-bit value that is derived from the IANA "Private Enterprise" ID.<br>Most significant four bits = Reserved (0000b).<br>000000h = Unspecified, 0FFFFFh = Reserved.<br>This value is binary encoded.<br>For example, the ID for the IPMI forum is 7154 decimal, which is 1BF2h, which will be stored in this record as F2h, 1Bh, and 00h for bytes 8 through 10, respectively. |
| **11-16** | OEM Defined | Defined according to the manufacturer identified by the Manufacturer ID field. |

## Table 4. OEM SEL Record (Type E0h-Ffh)

| Byte | Field | Description |
|---|---|---|
| **1, 2** | Record ID (RID) | ID used for SEL Record access. |
| **3** | Record Type (RT) | [7:0] – Record Type<br>E0h-FFh = OEM system event record |
| **4–16** | OEM | OEM Defined. This is defined by the system integrator. |

## 2.2 Notes on SEL Logs and Collecting SEL Information

When capturing the SEL log, always collect both the text/human readable version and the hex version. Because some of the data is OEM-specific, some utilities cannot decode the information correctly. In addition, with some OEM-specific data there may be additional variables that are not decoded at all.

The following section provides examples of not decoding all the information.

See section 2.2.1 for the PCIe* errors. The type of error and the PCI Bus, Device, and Function are all a part of Event Data 1 through Event Data 3.

See section 2.2.2 for the Power Supply events. When there is a failure, predictive failure, or a configuration error, Event Data 2 and Event Data 3 hold additional information describing the Power Supply PMBus* Command Registers and values for that particular event.

### 2.2.1 Example of Decoding a PCIe* Correctable Error Events

The following is an example of decoding a PCIe* correctable error event. For this particular event, it recorded a receiver error on Bus 0, Device 2, and Function 2. Correctable errors are acceptable and normal at a low rate of occurrence.

RID[27][00] RT[02] TS[0A][9B][2E][50] GID[33][00] ER[04] ST[13] SN[05] EDIR[71] ED1[A0] ED1[00] ED3[12]

RID (Record ID) = 0027h
RT (Record Type) = 02h = system event record
TS (Timestamp) = 502E9B0Ah
GID (Generator ID = 0033h = BIOS SMI Handler
ER (Event Message Revision) = 04h = IPMI v2.0
ST (Sensor Type) = 13h = Critical Interrupt (From *IPMI Specification* Table 42-3, Sensor Type Codes)
SN (Sensor Number = 05h
EDIR (Event Direction/Event Type) = 71h
[7] = 0b = Assertion Event
[6:0] = 71h = OEM Specific for PCI Express* correctable errors
ED1 (Event Data 1) = A0h
[7:6] = 10b = OEM code in Event Data 2
[5:4] – 10b = OEM code in Event Data 3
[3:0] – Event Trigger Offset = 0h = Receiver Error
ED2 (Event Data 2) = 00h; PCI Bus number = 0h
ED3 (Event Data 3) = 12h
[7:3] – PCI Device number = 02h
[2:0] – PCI Function number = 2h

### 2.2.2 Example of Decoding a Power Supply Predictive Failure Event

The following is an example of decoding a Power Supply predictive failure event. For this example, power supply 1 saw an AC power loss event with both the input under-voltage warning and fault events getting set. In most cases, this means that the AC power spiked under the minimum warning and fault thresholds for over 20 milliseconds but the system remained powered on. If these events continue to occur, it is advisable to check the power source.

RID[5D][00] RT[02] TS[D3][B1][AE][4E] GID[20][00] ER[04] ST[08] SN[50] EDIR[6F] ED1[A2] ED2[06] ED3[30]

RID (Record ID) = 005Dh
RT (Record Type) = 02h = system event record
TS (Timestamp) = 4EAEB1D3h
GID (Generator ID = 0020h = BMC
ER (Event Message Revision) = 04h = IPMI v2.0
ST (Sensor Type) = 08h = Power Supply (From *IPMI Specification* Table 42-3, Sensor Type Codes)
SN (Sensor Number = 50h = Power Supply 1

EDIR (Event Direction/Event Type) = 6Fh
- [7] = 0b = Assertion Event
- [6:0] = 6fh = Sensor specific

ED1 (Event Data 1) = A2h
- [7:6] = 10b = OEM code in Event Data 2
- [5:4] – 10b = OEM code in Event Data 3
- [3:0] – Event Trigger Offset = 2h = Predictive Failure

ED2 (Event Data 2) = 06h = Input under-voltage warning

ED3 (Event Data 3) = 30h; From *PMBus* Specification* STATUS_INPUT command
- [5] – VIN_UV_WARNING (Input Under-voltage Warning) = 1b
- [4] – VIN_UV_FAULT (Input Under-voltage Fault) = 1b

### 2.2.3 Example of Decoding an NVMe* Temperature Sensor Event

The following is an example of decoding an NVMe* temperature sensor event. The NVMe* temperature SEL occurs when the aggregate temperature sensor reading reaches the upper critical threshold. To report which of several drives are over temperature, OEM extended data bytes are used. The remainder of the SEL uses standard temperature sensor decoding rules.

The NVMe* temperature SEL uses a new format for the OEM extended data bytes. See the *Intel® Server System Integrated Baseboard Management Controller Firmware External Product Specification*.

ExtData[0] already has an established format. Bits 0–1 define the severity of the SEL. Bits 4–6 define the number of extended data bytes. Bits 2–3 and 7 are reserved. The new format is detailed below.

ExtData[0]
- [7] = Extended format indicator
- [6:4] = # Extended Data bytes
- [3:2] = reserved
- [1:0] = Existing severity encoding

ExtData[1] = Extended Format Byte Code
- 0 = Reserved
- 1 = Drive Position Format
- 2 = Critical Warning Format
- 3 = Add in card format
- 4-0xff = Reserved

ExtData[2]
- For Format = 1 (Drive Position Format), ExtData[2] is a bitfield indicating which drives are over temperature. Bit 0 set indicates Drive 0 is over temp. Bit 7 set indicates Drive 7 is over temp.

Exertion example:

RID:0099 RT:02 TS:558BEF64 GID:0020 ER:04 ST:01 S#:92 ET:01 ED:59 01 00 OEM:A2 01 20 FF FF FF FF FF
- ExtData[0] = 0xA2
  - Bit 7 set, so ExtData[1] is format indicator
  - Bit 6-4 = 2, indicating 2 more OEM bytes (format, bitfield)
  - Bit 1–0 = 2, indicating this is a Critical SEL
- ExtData[1] = 0x01, indicating Drive Position Format
- ExtData[2] = 0x20, indicating drive slot 5 (zero based) is over temp

# 3. Sensor Cross Reference List

This chapter contains a cross reference to help find details on any specific SEL entry.

## 3.1 BMC-Owned Sensors (GID = 0020h)

The following table can be used to find the details of sensors owned by the BMC.

**Table 5. M50CYP BMC-Owned Sensors**

| Sensor # | Sensor Name | Details Section | Next Steps |
|---|---|---|---|
| 01h | Power Unit Status (Pwr Unit Status) | 4.4.1 | Table 19 |
| 02h | Power Unit Redundancy (Pwr Unit Redund) | 4.4.2 | Table 21 |
| 03h | IPMI Watchdog (IPMI Watchdog) | 11.1 | Table 86 |
| 04h | Physical Security (Physical Scrty) | 10.1 | Table 82 |
| 05h | FP Interrupt (FP NMI Diag Int) | 10.2 | Section 10.2.1 |
| 06h | SMI Timeout (SMI Timeout) | 11.2 | Section 11.2.1 |
| 07h | System Event Log (System Event Log) | 11.3 | Not applicable |
| 08h | System Event (System Event) | 11.4 | Section 11.4.1 |
| 09h | Button Sensor (Button) | 10.3 | Not applicable |
| 0Ah | BMC Watchdog (BMC Watchdog) | 11.5 | Section 11.5.1 |
| 0Bh | Voltage Regulator Watchdog (VR Watchdog) | 4.3 | Section 4.3.1 |
| 0Ch | Fan Redundancy (Fan Redundancy) | 5.1.2 | Table 37 |
| 0Dh | SSB Thermal Trip (SSB Thermal Trip) | 5.2.6.1 | Section 5.2.6.3 |
| 0Eh | OCP Module Presence (OCP Mod Presence) | 11.7 | Section 11.7.1 |
| 0Fh | SAS Module Presence (SAS Mod Presence) | 11.7 | Section 11.7.1 |
| 10h | BMC Firmware Health (BMC FW Health) | 11.8 | Section 11.8.1 |
| 11h | System Airflow (System Airflow) | 5.3 | Not applicable |
| 12h | Firmware Update Status (FW Update Status) | 11.9 | Not applicable |
| 14h | Baseboard Temperature 5 (BB Lft Rear Temp) | 5.2.1 | Table 40 |
| 1Ah | Firmware Security(FW Security) | 11.13 | Not applicable |
| 20h | Baseboard Temperature 1 (BB P0 VR Temp) | 5.2.1 | Table 40 |
| 21h | Front Panel Temperature (Front Panel Temp) | 5.2.1 | Table 40 |
| 22h | PCH Temperature(PCH Temp) | 5.2.1 | Table 40 |
| 23h | Baseboard Temperature 2 (BB P1 VR Temp) | 5.2.1 | Table 40 |
| 24h | Baseboard Temperature 3 (BB BMC Temp) | 5.2.1 | Table 40 |
| 25h | Baseboard Temperature 4 (BB M.2 Temp) | 5.2.1 | Table 40 |
| 26h | OCP Module Temperature (OCP Mod Temp) | 5.2.1 | Table 40 |
| 27h | Hot-swap Backplane 1 Temperature (1U HSBP Temp) | 12.1 | Table 103 |
| 29h | Hot-swap Backplane 1 Temperature (2U HSBP 1 Temp) | 12.1 | Table 103 |
| 2Ah | Hot-swap Backplane 2 Temperature (2U HSBP 2 Temp) | 12.1 | Table 103 |
| 2Bh | Hot-swap Backplane 3 Temperature (2U HSBP 3 Temp) | 12.1 | Table 103 |
| 2Ch | PCI Riser 1 Temperature (Riser 1 Temp) | 5.2.1 | Table 40 |
| 2Dh | PCI Riser 2 Temperature (Riser 2 Temp) | 5.2.1 | Table 40 |
| 2Eh | Exit Air Temperature (Exit Air Temp) | 5.2.1 | Table 40 |
| 2Fh | PCI Riser 3 Temperature (Riser 3 Temp) | 5.2.1 | Table 40 |
| 30h-3Fh | Fan Tachometer Sensors (Chassis specific sensor names) | 5.1.1 | Table 33 |
| 40h-45h | Fan Present Sensors (Fan x Present) | 5.1.2 | Table 35 |

| Sensor # | Sensor Name | Details Section | Next Steps |
|---|---|---|---|
| 4Ch | Processor 1 VR controller temp (P0 VR Ctrl Temp) | 5.2.1 | Table 40 |
| 4Dh | Processor 2 VR controller temp (P1 VR Ctrl Temp) | 5.2.1 | Table 40 |
| 50h | Power Supply 1 Status (PS1 Status) | 4.5.1 | Table 19 |
| 51h | Power Supply 2 Status (PS2 Status) | 4.5.1 | Table 19 |
| 54h | Power Supply 1 AC Power Input (PS1 Input Power) | 4.5.2 | Table 26 |
| 55h | Power Supply 2 AC Power Input (PS2 Input Power) | 4.5.2 | Table 26 |
| 58h | Power Supply 1 +12V % of Maximum Current Output (PS1 Curr Out %) | 4.5.3 | Table 28 |
| 59h | Power Supply 2 +12V % of Maximum Current Output (PS2 Curr Out %) | 4.5.3 | Table 28 |
| 5Ch | Power Supply 1 Temperature (PS1 Temperature) | 4.5.4 | Table 30 |
| 5Dh | Power Supply 2 Temperature (PS2 Temperature) | 4.5.4 | Table 30 |
| 5Eh | PCIe* Aggregate Thermal Margin (PCIe* Aggr Margin) | 5.2.2 | Table 43 |
| 60h–68h | Hard Disk Drive 15–23 Status (HDD 15–23 Status) | 12.2 | Table 105 |
| 69h | Midplane 1 Temp (Left Swch Temp) | 5.2.1 | Table 40 |
| 6Ah | Midplane 2 Temp (Right Swch Temp) | 5.2.1 | Table 40 |
| 6Bh | SAS Module Temperature (SAS Mod Temp) | 5.2.1 | Table 40 |
| 70h | Processor 1 Status (P1 Status) | 6.1 | Table 57 |
| 71h | Processor 2 Status (P2 Status) | 6.1 | Table 57 |
| 78h | Processor 1 Thermal Control % (P0 D1 Thm Ctrl %) | 5.2.3 | Section 5.2.3.1 |
| 7Ah | Processor 2 Thermal Control % (P1 D1 Thm Ctrl %) | 5.2.3 | Section 5.2.3.1 |
| 7Ch | Processor ERR2 Timeout (CPU ERR2) | 6.5 | Section 6.5.1 |
| 81h | MTM Level Change (MTM Lvl Change) | 11.6 | Not applicable |
| 82h | Processor Population Fault (CPU Missing) | 6.3 | Section 6.3.1 |
| 83h | Processor 1 DTS Thermal Margin (P0 D1 DTS Th Mgn) | 5.2.4 | Not applicable |
| 85h | Processor 1 DTS Temperature (P0 D1 Abs Temp) | 5.2.1 | Table 40 |
| 87h | Auto Config Status (AutoCfg Status) | 11.10 | Section 11.10.1 |
| 88h | Processor 2 DTS Thermal Margin (P1 D1 DTS Th Mgn) | 5.2.4 | Not applicable |
| 8Ah | Processor 2 DTS Temperature (P1 D1 Abs Temp) | 5.2.1 | Table 40 |
| 90h | VRD Over Temperature (VRD Hot) | 5.2.5 | Section 5.2.6.3 |
| 91h | NVMe* 1 Therm Mgn | 5.2.9.1 | Section 5.2.9.1 |
| 92h | NVMe* 2 Therm Mgn | 5.2.9.1 | Section 5.2.9.1 |
| 93h | NVMe* 3 Therm Mgn | 5.2.9.1 | Section 5.2.9.1 |
| 94h | NVMe* 1 Crit Warn | 5.2.9.2 | Table 55 |
| 95h | NVMe* 2 Crit Warn | 5.2.9.2 | Table 55 |
| 96h | NVMe* 3 Crit Warn | 5.2.9.2 | Table 55 |
| A0h | Power Supply 1 Fan Fail[3] (PS1 Fan Fail) | 4.5.5 | Section 4.5.5.1 |
| A4h | Power Supply 2 Fan Fail[3] (PS2 Fan Fail) | 4.5.5 | Section 4.5.5.1 |
| B0h | Processor 1 DIMM Aggregate Thermal Margin 1 (DIMM Thrm Mrgn 1) | 5.2.2 | Table 43 |
| B1h | Processor 1 DIMM Aggregate Thermal Margin 2 (DIMM Thrm Mrgn 2) | 5.2.2 | Table 43 |
| B2h | Processor 2 DIMM Aggregate Thermal Margin 3 (DIMM Thrm Mrgn 3) | 5.2.2 | Table 43 |
| B3h | Processor 2 DIMM Aggregate Thermal Margin 4 (DIMM Thrm Mrgn 4) | 5.2.2 | Table 43 |
| C0h | Processor 1 DIMM Thermal Trip (Mem P1 Thrm Trip) | 5.2.8 | Section 5.2.8.1 |
| C1h | Processor 2 DIMM Thermal Trip (Mem P2 Thrm Trip) | 5.2.8 | Section 5.2.8.1 |
| C8h | Global Aggregate Temperature Margin 1 (Agg Therm Mgn 1) | 5.2.2 | Table 43 |
| C9h | Global Aggregate Temperature Margin 2 (Agg Therm Mgn 2) | 5.2.2 | Table 43 |
| D0h | Baseboard +12V Right (BB +12.0V_R) | 4.1 | Table 14 |
| D1h | Voltage Fault (Voltage Fault) | 4.2 | Table 16 |

| Sensor # | Sensor Name | Details Section | Next Steps |
|---|---|---|---|
| D7h | Bad User Password Entered (Bad User PWD) | 11.11 | Section 11.11 |
| DAh | KCS Policy | 11.14 | Section 11.14 |
| DBh | Remote Debug | 11.12 | Section 11.12 |
| DEh | Baseboard CMOS Battery (BB +3.3V Vbat) | 4.1 | Table 14 |
| E0h | Hot-swap Backplane 4 Temperature (HSBP 4 Temp) | 12.1 | Table 103 |
| E2h-E3h | Rear Hard Disk Drive 0 -1 Status (Rear HDD 0–1 Stat) | 12.2 | Table 105 |
| F0h-FEh | Hard Disk Drive 0–14 Status (HDD 0–14 Status) | 12.2 | Table 105 |

1. Redundancy sensors will be only present on systems with appropriate hardware to support redundancy (for instance, fan or power supply). Note that power supply redundancy may be lost even when both supplies are operational if the system is loaded beyond the capacity of a single power supply.
2. This is only applicable when the system doesn't support redundant fans. When fan redundancy is supported, then the contribution to system state is driven by the fan redundancy sensor, not individual sensors. On a system with fan redundancy, the individual sensor severities will read the same as the fan redundancy sensor's severity.
3. This is only applicable when the system doesn't support redundant power supplies. When redundancy is supported, then the contribution to system state is driven by the power unit redundancy sensor. On a system with power supply redundancy, the individual sensor severities will read the same as the power unit redundancy sensor's severity.

## Table 6. D50TNP BMC

| Sensor # | Sensor Name | Details Section | Next Steps |
|---|---|---|---|
| 01h | Power Unit Status (Pwr Unit Status) | 4.4.1 | Table 19 |
| 02h | Power Unit Redundancy (Pwr Unit Redund) | 4.4.2 | Table 21 |
| 03h | IPMI Watchdog (IPMI Watchdog) | 11.1 | Table 86 |
| 04h | Physical Security (Physical Scrty) | 10.1 | Table 82 |
| 05h | FP Interrupt (FP NMI Diag Int) | 10.2 | Section 10.2.1 |
| 06h | SMI Timeout (SMI Timeout) | 11.2 | Section 11.2.1 |
| 07h | System Event Log (System Event Log) | 11.3 | Not applicable |
| 08h | System Event (System Event) | 11.4 | Section 11.4.1 |
| 09h | Button Sensor (Button) | 10.3 | Not applicable |
| 0Ah | BMC Watchdog (BMC Watchdog) | 11.5 | Section 11.5.1 |
| 0Bh | Voltage Regulator Watchdog (VR Watchdog) | 4.3 | Section 4.3.1 |
| 0Dh | PCH Thermal Trip (PCH Therm Trip) | 5.2.6.1 | Section 5.2.6.3 |
| 10h | BMC Firmware Health (BMC FW Health) | 11.7 | Section 11.8.1 |
| 11h | System Airflow (System Airflow) | 5.3 | Not applicable |
| 12h | Firmware Update Status (FW Update Status) | 11.9 | Not applicable |
| 14h | Baseboard Temperature 5 (BB Outlet Temp) | 5.2.1 | Table 40 |
| 1Ah | Firmware Security(FW Security) | 11.13 | Not applicable |
| 21h | Baseboard Temperature 2 (BB inlet Temp) | 5.2.1 | Table 40 |
| 22h | PCH Temperature (PCH Temp) | 5.2.1 | Table 40 |
| 24h | Baseboard Temperature3 (CPU0 VR Temp) | 5.2.1 | Table 40 |
| 25h | Network Interface Controller Temperature (LAN Temp) | 5.2.1 | Table 40 |
| 26h | Baseboard Temperature (CPU1 VR Temp) | 5.2.1 | Table 40 |
| 27h | PCI Riser 1 Temperature (Riser 1 Temp) | 5.2.1 | Table 40 |
| 28h | U.2 NVMe* Temperature (U.2 SSD Temp1) | 5.2.1 | Table 40 |
| 29h | M.2 NVMe* Temperature (M.2 Temp1) | 5.2.1 | Table 40 |
| 2Ah | PDB Temperature (PDB Temperature) | 5.2.1 | Table 40 |
| 2Ch | PCI Riser 2 Temperature (Riser 2 Temp) | 5.2.1 | Table 40 |
| 2Dh | U.2 NVMe* Temperature (U.2 SSD Temp2) | 5.2.1 | Table 40 |

| Sensor # | Sensor Name | Details Section | Next Steps |
|---|---|---|---|
| 2Eh | Exit Air Temperature (Exit Air Temp) | 5.2.1 | Table 40 |
| 2Fh | Network Interface Controller Temperature (LAN NIC Temp) | 5.2.1 | Table 40 |
| 30h | Fan Tachometer Sensors[2] (Sys Fan 1A) | 5.1.1 | Table 33 |
| 30h | Fan Tachometer Sensors[2] (Sys Fan 4A) | 5.1.1 | Table 33 |
| 31h | Fan Tachometer Sensors[2] (Sys Fan 1B) | 5.1.1 | Table 33 |
| 31h | Fan Tachometer Sensors[2] (Sys Fan 4B) | 5.1.1 | Table 33 |
| 32h | Fan Tachometer Sensors[2] (Sys Fan 2A) | 5.1.1 | Table 33 |
| 32h | Fan Tachometer Sensors[2] (Sys Fan 5A) | 5.1.1 | Table 33 |
| 33h | Fan Tachometer Sensors[2] (Sys Fan 2B) | 5.1.1 | Table 33 |
| 33h | Fan Tachometer Sensors[2] (Sys Fan 5B) | 5.1.1 | Table 33 |
| 34h | Fan Tachometer Sensors[2] (Sys Fan 3A) | 5.1.1 | Table 33 |
| 34h | Fan Tachometer Sensors[2] (Sys Fan 6A) | 5.1.1 | Table 33 |
| 35h | Fan Tachometer Sensors[2] (Sys Fan 3B) | 5.1.1 | Table 33 |
| 35h | Fan Tachometer Sensors[2] (Sys Fan 6B) | 5.1.1 | Table 33 |
| 36h | Fan Tachometer Sensors[2] (Sys Fan 6) | 5.1.1 | Table 33 |
| 37h | Fan Tachometer Sensors[2] (Sys Fan 7) | 5.1.1 | Table 33 |
| 44h | EV CPU 1 VR Temperature (EV CPU1VR Temp) | 5.2.1 | Table 40 |
| 45h | Memory 0 VRD Temperature (Mem 0 VRD Temp) | 5.2.1 | Table 40 |
| 46h | Memory 1 VRD Temperature (Mem 1 VRD Temp) | 5.2.1 | Table 40 |
| 47h | M.2 NVMe* Temperature (M.2 Temp 2) | 5.2.1 | Table 40 |
| 50h | Power Supply 1 Status (PS1 Status) | 4.5.1 | Table 19 |
| 51h | Power Supply 2 Status (PS2 Status) | 4.5.1 | Table 19 |
| 54h | Power Supply 1 AC Power Input (PS1 Input Power) | 4.5.2 | Table 26 |
| 55h | Power Supply 2 AC Power Input (PS2 Input Power) | 4.5.2 | Table 26 |
| 58h | Power Supply 1 +12V % of Maximum Current Output (PS1 Curr Out %) | 4.5.3 | Table 28 |
| 59h | Power Supply 2 +12V % of Maximum Current Output (PS2 Curr Out %) | 4.5.3 | Table 28 |
| 5Ch | Power Supply 1 Temperature (PS1 Temperature) | 4.5.4 | Table 30 |
| 5Dh | Power Supply 2 Temperature (PS2 Temperature) | 4.5.4 | Table 30 |
| 60h | EDSFF Dock Temperature (Dock Temp) | 5.2.1 | Table 40 |
| 61h | Max RSSD temperature (RSSD Aggr Thrm 1) | 5.2.1 | Table 40 |
| 62h | Max RSSD temperature (RSSD Aggr Thrm 2) | 5.2.1 | Table 40 |
| 70h | Processor 1 Status (P1 Status) | 6.1 | Table 57 |
| 71h | Processor 2 Status (P2 Status) | 6.1 | Table 57 |
| 74h | Processor 1 Thermal Margin (P1 Therm Margin) | 5.2.2 | Table 43 |
| 75h | Processor 2 Thermal Margin (P2 Therm Margin) | 5.2.2 | Table 43 |
| 78h | Processor 1 Thermal Control % (P1 Therm Ctrl %) | 5.2.3 | Section 5.2.3.1 |
| 79h | Processor 2 Thermal Control % (P2 Therm Ctrl %) | 5.2.3 | Section 5.2.3.1 |
| 7Ch | Processor ERR2 Timeout (CPU ERR2) | 6.5 | Section 6.5.1 |
| 81h | MTM Level Change (MTM Lvl Change) | 11.6 | Not applicable |
| 82h | Processor Population Fault (CPU Missing) | 6.3 | Section 6.3.1 |
| 83h–84h | Processor 1–2 DTS Thermal Margin (P1-P2 DTS Therm Mgn) | 5.2.4 | Not applicable |
| 85h–86h | Processor 1–2 DTS Temperature (P1-P2 DTS Temperature) | 5.2.1 | Table 40 |
| 87h | Auto Config Status (AutoCfg Status) | 11.10 | Section 11.10.1 |
| 88h | GPGPU Riser Temperature (GPGPU Riser) | 5.2.1 | Table 40 |
| 90h | VRD Over Temperature (VRD Hot) | 5.2.5 | Section 5.2.6.3 |
| 91h | NVMe* 1 Therm Mgn | 5.2.9.1 | Section 5.2.9.1 |

| Sensor # | Sensor Name | Details Section | Next Steps |
|---|---|---|---|
| 93h | CPU0 VCCANA Temp | 5.2.1 | Table 40 |
| 94h | CPU1 VCCANA Temp | 5.2.1 | Table 40 |
| A0h | Power Supply 1 Fan Fail[3] (PS1 Fan1 Fail) | 4.5.5 | Section 4.5.5.1 |
| A1h | Power Supply 1 Fan 2 Fail[3] (PS1 Fan2 Fail) | 4.5.5 | Section 4.5.5.1 |
| A4h | Power Supply 2 Fan Fail[3] (PS2 Fan1 Fail) | 4.5.5 | Section 4.5.5.1 |
| A5h | Power Supply 2 Fan 2 Fail[3] (PS2 Fan2 Fail) | 4.5.5 | Section 4.5.5.1 |
| B0h | Processor 1DIMM Aggregate Thermal Margin 1 (P1 DIMM Thrm 1) | 5.2.2 | Table 43 |
| B1h | Processor 1 DIMM Aggregate Thermal Margin 2 (P1 DIMM Thrm 2) | 5.2.2 | Table 43 |
| B2h | Processor 2 DIMM Aggregate Thermal Margin 1 (P2 DIMM Thrm 1) | 5.2.2 | Table 43 |
| B3h | Processor 2 DIMM Aggregate Thermal Margin 2 (P2 DIMM Thrm 2) | 5.2.2 | Table 43 |
| B8h | Node Auto-Shutdown Sensor (Auto Shutdown) | 4.4.3 | 4.4.3.1 |
| C0h | Processor 1 DIMM Thermal Trip (Mem P1 Thrm Trip) | 5.2.8 | Section 5.2.8.1 |
| C1h | Processor 2 DIMM Thermal Trip (Mem P2 Thrm Trip) | 5.2.8 | Section 5.2.8.1 |
| C8h | Global Aggregate Temperature Margin 1 (Agg Therm Mgn 1) | 5.2.2 | Table 43 |
| C9h | Global Aggregate Temperature Margin 2 (Agg Therm Mgn 2) | 5.2.2 | Table 43 |
| D0h | Baseboard +12V Right (BB +12.0V_R) | 4.1 | Table 14 |
| D1h | Voltage Fault (Voltage Fault) | 4.2 | Table 16 |
| D7h | Bad User Password Entered (Bad User PWD) | 11.11 | Section 11.11 |
| DAh | KCS Policy | 11.14 | Section 11.14 |
| DBh | Remote Debug | 11.12 | Section 11.12 |
| DEh | Baseboard CMOS Battery (BB +3.3V Vbat) | 4.1 | Table 14 |
| F0h-F5h | Hard Disk Drive 0–5 Status (HDD 0–5 Status) | 12.2 | Table 105 |

1. Redundancy sensors will be only present on systems with appropriate hardware to support redundancy (for instance, fan or power supply). Note that power supply redundancy may be lost even when both supplies are operational if the system is loaded beyond the capacity of a single power supply.
2. This is only applicable when the system doesn't support redundant fans. When fan redundancy is supported, then the contribution to system state is driven by the fan redundancy sensor, not individual sensors. On a system with fan redundancy, the individual sensor severities will read the same as the fan redundancy sensor's severity.
3. This is only applicable when the system doesn't support redundant power supplies. When redundancy is supported, then the contribution to system state is driven by the power unit redundancy sensor. On a system with power supply redundancy, the individual sensor severities will read the same as the power unit redundancy sensor's severity.

## 3.2 BIOS POST-Owned Sensors (GID = 0001h)

The following table can be used to find the details of sensors owned by BIOS POST.

**Table 7. BIOS POST Owned Sensors**

| Sensor # | Sensor Name | Details Section | Next Steps |
|---|---|---|---|
| 02h | Memory RAS Configuration Status/Memory ECC Error | 7.1 | Table 64 |
| 06h | BIOS POST Error | 9.2 | Section 9.2.1 |
| 09h | Intel® UPI Link Width Reduced | 6.4.1 | Section 6.4.1.1 |
| 12h | Memory RAS Mode Select | 7.2 | Not applicable |
| 73h | OOB Firmware update/OOB BIOS Configuration | | |
| 83h | System Boot Event | | |

## 3.3 BIOS SMI Handler-Owned Sensors (GID = 0033h)

The following table can be used to find the details of sensors owned by BIOS System Management Interrupt (SMI) Handler.

**Table 8. BIOS SMI Handler Owned Sensors**

| Sensor # | Sensor Name | Details Section | Next Steps |
|---|---|---|---|
| 01h | Mirroring Redundancy State | 7.3 | Section 7.3.1 |
| 02h | Memory ECC Error | 7.6.1 | Table 71 |
| 03h | Legacy PCI Error | 8.1 | Section 8.1.1 |
| 04h | PCI Express* Fatal Error (Standard AER Errors) (see Sensor 14h for continuation) | 8.2.1 | Section 8.2.1.1 |
| 05h | PCI Express* Correctable Error (Standard AER Errors) | 8.2.2 | Section 8.2.2.1 |
| 06h | Intel UPI Correctable Error(reserved for validation) | 6.4.2 | Section 6.4.2.1 |
| 08h | IIO Internal Error | | |
| 10h | Memory Error Extension(reserved for validation) | | |
| 11h | Spring Redundancy State | 7.4 | 7.4.1 |
| 12h | Memory RAS Mode select | 7.2 | Section 7.2 |
| 13h | Memory Parity Error | 7.6.2 | Section 7.6.2.1 |
| 14h | PCI Express* Fatal Error#2 (continuation of Sensor 04h) | 8.2.1 | Section 8.2.1.1 |
| 20h | ADDDC Error | 7.5 | Section 7.5 |

## 3.4 Intel® NM/Intel® ME Firmware-Owned Sensors (GID = 002Ch or 602Ch)

The following table can be used to find the details of sensors owned by the Intel® Node Manager (Intel® NM)/Intel® Management Engine (Intel® ME) firmware.

**Table 9. Intel® Management Engine Firmware-Owned Sensors**

| Sensor # | Sensor Name | Details Section | Next Steps |
|---|---|---|---|
| 17h | Intel® ME Firmware Health Events | 13.1 | Section 13.1.1 |
| 18h | Intel® Node Manager Exception Events | 13.2 | Section 13.2.1 |
| 19h | Intel® Node Manager Health Events | 13.3 | Section 13.3.1 |
| 1Ah | Intel® Node Manager Operational Capabilities Change Events | 13.4 | Section 13.4.1 |
| 1Bh | Intel® Node Manager Alert Threshold Exceeded Events | 13.5 | Section 13.5.1 |
| B2h | Intel® Node Manager SmaRT/CLST Events | 13.6 | Section 13.6.1 |

## 3.5 Microsoft* OS-Owned Events (GID = 0041h)

The following table can be used to find the details of records that are owned by the Microsoft* Operating System (OS).

**Table 10. Microsoft* OS-Owned Events**

| Sensor Name | Record Type | Sensor Type | Details Section | Next Steps |
|---|---|---|---|---|
| Boot Event | 02h | 1Fh = OS Boot | Table 114 | Not applicable |
| | DCh | Not applicable | Table 115 | Not applicable |
| Shutdown Event | 02h | 20h = OS Stop/Shutdown | Table 116 | Not applicable |
| | DDh | Not applicable | Table 117, Table 118 | Not applicable |
| | 02h | 20h = OS Stop/Shutdown | Table 119 | Not applicable |

| Sensor Name | Record Type | Sensor Type | Details Section | Next Steps |
|---|---|---|---|---|
| **Bug Check/ Blue Screen** | DEh | Not applicable | Table 120 | Not applicable. |

## 3.6 Linux* Kernel Panic Events (GID = 0021h)

The following table can be used to find the details of records that can be generated when there is a Linux* Kernel panic.

**Table 11. Linux* Kernel Panic Events**

| Sensor Name | Record Type | Sensor Type | Details Section | Next Steps |
|---|---|---|---|---|
| **Linux* Kernel Panic** | 02h | 20h = OS Stop/Shutdown | Table 121 | Not applicable |
| | F0h | Not applicable | Table 122 | Not applicable |

# 4. Power Subsystems

The BMC monitors the power subsystem including power supplies, select onboard voltages, and related sensors.

## 4.1 Threshold-Based Voltage Sensors

The BMC monitors the main voltage sources in the system, including the baseboard, memory, and processors, using IPMI-compliant analog/threshold sensors. Some voltages are only on specific platforms. For details check the platform *Technical Product Specification (TPS)*.

**Note**: A voltage error can be caused by the device supplying the voltage or by the device using the voltage. For each sensor, it is noted who is supplying the voltage and who is using it.

**Table 12. Threshold-Based Voltage Sensors Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 02h = Voltage |
| **12** | Sensor Number | See Table 14 |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 01h (Threshold) |
| **14** | Event Data 1 | [7:6] – 01b = Trigger reading in Event Data 2<br>[5:4] – 01b = Trigger threshold in Event Data 3<br>[3:0] – Event Triggers as described in Table 13 |
| **15** | Event Data 2 | Reading that triggered event, FFh or not present if unspecified. |
| **16** | Event Data 3 | Threshold value that triggered event, FFh or not present if unspecified. If present, byte 2 must be present. |

The following table describes the severity of each of the event triggers for both assertion and deassertion.

**Table 13. Threshold-Based Voltage Sensors Event Triggers**

| Event Trigger | | Assertion Severity | Deassertion Severity | Description |
|---|---|---|---|---|
| **00h** | Lower non-critical going low | Degraded | OK | The voltage has dropped below its lower non-critical threshold. |
| **02h** | Lower critical going low | Non-fatal | Degraded | The voltage has dropped below its lower critical threshold. |
| **07h** | Upper non-critical going high | Degraded | OK | The voltage has gone over its upper non-critical threshold. |
| **09h** | Upper critical going high | Non-fatal | Degraded | The voltage has gone over its upper critical threshold. |

**Table 14. Threshold-Based Voltage Sensors – Next Steps**

| Sensor # | Sensor Name | Next Steps |
|---|---|---|
| 0Bh | Voltage Regulator Watchdog (VR Watchdog) | |
| D0h | Baseboard +12V (BB +12.0V) | +12 V is supplied by the power supplies.<br>+12 V is used by SATA drives, fans, and PCI-E cards. In addition, it is used to generate various processor voltages.<br>1. Ensure all cables are connected correctly.<br>2. Check connections on the fans and HDDs.<br>3. If the issue follows the component, swap it, otherwise, replace the board.<br>4. If the issue remains, replace the power supplies. |
| D1h | Voltage Fault (Voltage Fault) | The voltages in this sensor are all derived on the server board from the 12 V or 3.3 V power.<br>1. Remove all but minimum components for operation and check sensor.<br>2. Inspect for contamination in connectors (DIMMS, PCIe*).<br>3. If error remains, replace the board. |
| DEh | Baseboard CMOS Battery (BB +3.3V Vbat) | Vbat is supplied by the CMOS battery when AC power is off (Battery Voltage is 3.0 V) and by the main board when AC power is on (Battery Voltage is 3.3 V).<br>Vbat is used by the CMOS and related circuits.<br>1. Replace the CMOS battery. Any battery of type CR2032 can be used.<br>2. If error remains (unlikely), replace the board. |

## 4.2 Aggregate Voltage Fault Sensor

The discrete voltage sensor monitors multiple voltages from sensors around the baseboard and then asserts a bit in the SEL event data for each sensor that is out of range. The Voltage name for the asserted bit can be retrieved using the Get Voltage Name IPMI function. For details, check *BMC External Product Specification (EPS)*.

**Table 15. Aggregate Voltage Sensors Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| 11 | Sensor Type | 02h = Voltage |
| 12 | Sensor Number | D1h |
| 13 | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 03h (Digital Discrete) |
| 14 | Event Data 1 | [7:6] – 10b = OEM code in Byte2<br>[5:4] – 10b = unspecified byte 3<br>[3:0] – 01b = Offset from Event/Reading Code for discrete event state |
| 15 | Event Data 2 | Voltage Fault Status<br>[Bit 0] = Not Used<br>[Bit 1] = +3.3V<br>[Bit 2] = VNN PCH Aux<br>[Bit 3] = +1.05V PCH Aux<br>[Bit 4] = +0.83V LAN Aux<br>[Bit 5] = 12V Aux<br>[Bit 6] = +1.8V PCH Aux<br>[Bit 7] = VCC In CPU0 |

| Byte | Field | Description |
|---|---|---|
| **16** | Event Data 3 | Voltage Fault Status<br>[Bit 0] = VCC In CPU1<br>[Bit 1] = VDDQ ABC CPU0<br>[Bit 2] = VDDQ DEF CPU0<br>[Bit 3] = VDDQ ABC CPU1<br>[Bit 4] = VDDQ DEF CPU1<br>[Bit 5] = VCCIO CPU0<br>[Bit 6] = VCCIO CPU1<br>[Bit 7] = Not Used |
| **17** | Extended Data 1 | [7] - Reserved<br>[6:4] - 110b = Extended SEL record length of 6 bytes<br>[3:2] - Reserved<br>[1:0] - Severity Status<br>10b = Severity Critical |
| **18** | Extended Data 2 | Voltage Fault Status = Not Used |
| **19** | Extended Data 3 | Voltage Fault Status = Not Used |
| **20** | Extended Data 4 | Upper Non-Critical Voltage Fault Status<br>[Bit 0] = Not Used<br>[Bit 1] = +3.3V<br>[Bit 2] = VNN PCH Aux<br>[Bit 3] = +1.05V PCH Aux<br>[Bit 4] = +0.83V LAN Aux<br>[Bit 5] = 12V Aux<br>[Bit 6] = +1.8V PCH Aux<br>[Bit 7] = VCC In CPU0 |
| **21** | Extended Data 5 | Upper Non-Critical Voltage Fault Status<br>[Bit 0] = VCC In CPU1<br>[Bit 1] = VDDQ ABC CPU0<br>[Bit 2] = VDDQ DEF CPU0<br>[Bit 3] = VDDQ ABC CPU1<br>[Bit 4] = VDDQ DEF CPU1<br>[Bit 5] = VCCIO CPU0<br>[Bit 6] = VCCIO CPU1<br>[Bit 7] = Not Used |
| **22** | Extended Data 6 | Lower Non-Critical Voltage Fault Status<br>[Bit 0] = Not Used<br>[Bit 1] = +3.3V<br>[Bit 2] = VNN PCH Aux<br>[Bit 3] = +1.05V PCH Aux<br>[Bit 4] = +0.83V LAN Aux<br>[Bit 5] = 12V Aux<br>[Bit 6] = +1.8V PCH Aux<br>[Bit 7] = VCC In CPU0 |
| **23** | Extended Data 7 | Lower Non-Critical Voltage Fault Status<br>[Bit 0] = VCC In CPU1<br>[Bit 1] = VDDQ ABC CPU0<br>[Bit 2] = VDDQ DEF CPU0<br>[Bit 3] = VDDQ ABC CPU1<br>[Bit 4] = VDDQ DEF CPU1<br>[Bit 5] = VCCIO CPU0<br>[Bit 6] = VCCIO CPU1<br>[Bit 7] = Not Used |
| **24** | Extended Data 8 | Not used `0xFF` |

## Table 16. Discrete Voltage Sensors – Next Steps

| Voltage Name | Next Steps |
|---|---|
| **BB 1.8V P1 VCC** | This 1.8 V line is supplied by the main board.<br>This 1.8 V line is used by processor 1.<br>1. Ensure all cables are connected correctly.<br>2. Check the processor is seated properly.<br>3. Cross test the processors. If the issue remains with the processor socket, replace the main board, otherwise the processor. |
| **BB 1.8V P2 VCC** | This 1.8 V line is supplied by the main board.<br>This 1.8 V line is used by processor 2.<br>1. Ensure all cables are connected correctly.<br>2. Check the processor is seated properly.<br>3. Cross test the processors. If the issue remains with the processor socket, replace the main board, otherwise the processor. |
| **BB 1.2V P1DDR AB** | This 1.2 V line is supplied by the main board.<br>This 1.2 V line is used by processor 1 memory slots A and B.<br>1. Ensure all cables are connected correctly.<br>2. Check the DIMMs are seated properly.<br>3. Cross test the DIMMs. If the issue remains with the DIMMs on this socket, replace the main board, otherwise the DIMM. |
| **BB 1.2V P1DDR CD** | This 1.2 V line is supplied by the main board.<br>This 1.2 V line is used by processor 1 memory slots C and D.<br>1. Ensure all cables are connected correctly.<br>2. Check the DIMMs are seated properly.<br>3. Cross test the DIMMs. If the issue remains with the DIMMs on this socket, replace the main board, otherwise the DIMM. |
| **BB 1.2V P2DDR EF** | This 1.2 V line is supplied by the main board.<br>This 1.2 V line is used by processor 2 memory slots E and F.<br>1. Ensure all cables are connected correctly.<br>2. Check the DIMMs are seated properly.<br>3. Cross test the DIMMs. If the issue remains with the DIMMs on this socket, replace the main board, otherwise the DIMM. |
| **BB 1.2V P2DDR GH** | This 1.2 V line is supplied by the main board.<br>This 1.2 V line is used by processor 2 memory slots G and H.<br>1. Ensure all cables are connected correctly.<br>2. Check the DIMMs are seated properly.<br>3. Cross test the DIMMs. If the issue remains with the DIMMs on this socket, replace the main board, otherwise the DIMM. |
| **BB 2.62V VPP** | This 2.62 V line is supplied by the main board.<br>This 2.62 V line is used by memory.<br>1. Ensure all cables are connected correctly.<br>2. Check the DIMMs are seated properly.<br>3. Cross test the DIMMs. If the issue remains with the DIMMs on special socket, replace the main board, otherwise the DIMM. |
| **BB 1.05V VCC IO** | This 1.05 V line is supplied by the main board.<br>This 1.05 V line is used by processor and PECI.<br>1. Ensure all cables are connected correctly.<br>2. Check the processors are seated properly.<br>3. Cross test the processors. If the issue remains with the processor on special socket, replace the main board, otherwise the Processor. |

| Voltage Name | Next Steps |
|---|---|
| **BB 1.5V PCH** | This 1.5 V line is supplied by the main board.<br>This 1.5 V line is used by PCH.<br>1. Ensure all cables are connected correctly.<br>2. If the issue remains, replace the board. |
| **BB 1.05V PCH** | This 1.05 V line is supplied by the main board.<br>This 1.05 V line is used by PCH.<br>1. Ensure all cables are connected correctly.<br>2. If the issue remains, replace the board. |
| **BB 1.05V ASW PCH** | This 1.05 V line is supplied by the main board.<br>This 1.05 V line is used by PCH.<br>1. Ensure all cables are connected correctly.<br>2. If the issue remains, replace the board. |
| **BB 0.9V IB CORE** | +0.9 V Core IB is supplied by the main board on specific platforms.<br>+0.9 V Core IB is used by the on-board Infiniband* controller on those specific platforms.<br>1. Ensure all cables are connected correctly.<br>2. If the issue remains, replace the board.<br>3. If the issue remains, replace the power supplies. |
| **BB 1.0V AUX LAN** | +1.0 V AUX LAN is supplied by the main board.<br>+1.0 V AUX LAN is used by on-board NIC Intel® X540.<br>1. Ensure all cables are connected correctly.<br>2. If the issue remains, replace the board.<br>3. If the issue remains, replace the power supplies. |
| **BB 5V AUX** | This 5 V line is supplied by the HSBP (Platform Specific).<br>This 5 V line is used optionally and depends on customer's design.<br>1. Ensure all cables are connected correctly.<br>2. If the issue remains, replace the board. |
| **BB 3.3V AUX** | +3.3 V AUX is supplied by the main board.<br>+3.3 V AUX is used by the BMC, clock chips, PCI-E Slot, on-board NIC, PCH.<br>1. Ensure all cables are connected correctly.<br>2. If the issue remains, replace the board.<br>3. If the issue remains, replace the power supplies. |
| **BB 1.0V AUX** | +1.0 V AUX is supplied by the main board.<br>+1.0 V AUX is used by the BMC, on-board NIC.<br>1. Ensure all cables are connected correctly.<br>2. If the issue remains, replace the board.<br>3. If the issue remains, replace the power supplies. |
| **BB 12V AUX** | +12 V AUX is supplied by the Power Supply.<br>+12 V AUX is used by the BMC, clock chips, PCIe* Slot, on-board NIC, PCH.<br>1. Ensure all cables are connected correctly.<br>2. If the issue remains, replace the board.<br>3. If the issue remains, replace the power supplies. |
| **BB 3.3V** | +3.3 V is supplied by the power supplies for pedestal systems, and supplied by the main board on rack-optimized systems.<br>+3.3 V is used by the PCIe* and PCI-X slots.<br>1. Ensure all cables are connected correctly.<br>2. Reseat any PCI-E cards.<br>3. Try PCI-E cards in other PCI-E slots.<br>4. If the issue follows the card, swap it; otherwise, replace the main board.<br>5. If the issue remains, replace the power supplies. |

| Voltage Name | Next Steps |
|---|---|
| **VNN PCH Aux** | This VNN PCH Aux line is supplied by the main board.<br>This VNN PCH Aux line is used by PCH.<br>1. Ensure all cables are connected correctly.<br>2. If the issue remains, replace the board. |
| **BB 1.05V PCH Aux** | This 1.05V Aux line is supplied by the main board.<br>This 1.05V Aux line is used by PCH.<br>1. Ensure all cables are connected correctly.<br>2. If the issue remains, replace the board. |
| **BB 0.83V LAN Aux** | 0.83V LAN AUX is supplied by the main board.<br>0.83V LAN AUX is used by on-board NIC controller<br>1. Ensure all cables are connected correctly.<br>2. If the issue remains, replace the board.<br>3. If the issue remains, replace the power supplies. |
| **BB 1.8V PCH Aux** | This 1.8V Aux line is supplied by the main board.<br>This 1.8V Aux line is used by PCH.<br>1. Ensure all cables are connected correctly.<br>2. If the issue remains, replace the board. |
| **VCC In CPU0** | This 1.8 V line is supplied by the main board.<br>This 1.8 V line is used by processor 1.<br>1. Ensure all cables are connected correctly.<br>2. Check the processor is seated properly.<br>3. Cross test the processors. If the issue remains with the processor socket, replace the main board, otherwise the processor. |
| **VCC In CPU1** | This 1.8 V line is supplied by the main board.<br>This 1.8 V line is used by processor 2.<br>1. Ensure all cables are connected correctly.<br>2. Check the processor is seated properly.<br>3. Cross test the processors. If the issue remains with the processor socket, replace the main board, otherwise the processor. |
| **VDDQ ABC CPU0** | This 1.2 V line is supplied by the main board.<br>This 1.2 V line is used by processor 1 memory slots A/B/C.<br>1. Ensure all cables are connected correctly.<br>2. Check the DIMMs are seated properly.<br>3. Cross test the DIMMs. If the issue remains with the DIMMs on this socket, replace the main board, otherwise the DIMM. |
| **VDDQ DEF CPU0** | This 1.2 V line is supplied by the main board.<br>This 1.2 V line is used by processor 1 memory slots D/E/F.<br>1. Ensure all cables are connected correctly.<br>2. Check the DIMMs are seated properly.<br>3. Cross test the DIMMs. If the issue remains with the DIMMs on this socket, replace the main board, otherwise the DIMM. |
| **VDDQ ABC CPU1** | This 1.2 V line is supplied by the main board.<br>This 1.2 V line is used by processor 2 memory slots A/B/C.<br>1. Ensure all cables are connected correctly.<br>2. Check the DIMMs are seated properly.<br>3. Cross test the DIMMs. If the issue remains with the DIMMs on this socket, replace the main board, otherwise the DIMM. |
| **VDDQ DEF CPU1** | This 1.2 V line is supplied by the main board.<br>This 1.2 V line is used by processor 2 memory slots D/E/F.<br>1. Ensure all cables are connected correctly.<br>2. Check the DIMMs are seated properly.<br>3. Cross test the DIMMs. If the issue remains with the DIMMs on this socket, replace the main board, otherwise the DIMM. |

| Voltage Name | Next Steps |
|---|---|
| **VCCIO CPU0** | This 1.0 V line is supplied by the main board.<br>This 1.0 V line is used by processor and PECI.<br>1. Ensure all cables are connected correctly.<br>2. Check the processors are seated properly.<br>3. Cross test the processors. If the issue remains with the processor on special socket, replace the main board, otherwise the Processor. |
| **VCCIO CPU1** | This 1.0 V line is supplied by the main board.<br>This 1.0 V line is used by processor and PECI.<br>1. Ensure all cables are connected correctly.<br>2. Check the processors are seated properly.<br>3. Cross test the processors. If the issue remains with the processor on special socket, replace the main board, otherwise the Processor. |

## 4.3 Voltage Regulator Watchdog Timer Sensor

The BMC firmware monitors that the power sequence for the board voltage regulator (VR) controllers is completed when a DC power-on is initiated. Incompletion of the sequence indicates a board problem, in which case the firmware powers down the system.

The sequence is as follows:

- BMC firmware monitors the `PowerSupplyPowerGood` signal for assertion, indicating a DC-power-on has been initiated, and starts a timer (VR Watchdog Timer). For Intel Server Systems supporting the Intel® Xeon® Scalable processor family, this timeout is one second.
- If the `SystemPowerGood` signal has not asserted by the time the VR Watchdog Timer expires, then the firmware powers down the system, logs a SEL entry, and emits a beep code (1-5-1-2). This failure is termed as VR Watchdog Timeout.

**Table 17. Voltage Regulator Watchdog Timer Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | `02h` = Voltage |
| **12** | Sensor Number | `0Bh` |
| **13** | Event Direction and Event Type | [7] Event direction<br>`0b` = Assertion Event<br>`1b` = Deassertion Event<br>[6:0] Event Type = 03h (“digital” Discrete) |
| **14** | Event Data 1 | [7:6] – `00b` = Unspecified Event Data 2<br>[5:4] – `00b` = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset = `1h` = State Asserted |
| **15** | Event Data 2 | Not used |
| **16** | Event Data 3 | Not used |

### 4.3.1 Voltage Regulator Watchdog Timer Sensor – Next Steps

1. Ensure that all the connectors from the power supply are well seated.
2. Cross test the baseboard. If the issue remains with the baseboard, replace the baseboard.

## 4.4 Power Unit

The power unit monitors the power state of the system and logs the state changes in the SEL.

### 4.4.1 Power Unit Status Sensor

The power unit status sensor monitors the power state of the system and logs state changes. Expected power-on events, such as DC ON/OFF, are logged and unexpected events, such as AC loss and power good loss, are also logged.

**Table 18. Power Unit Status Sensors Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 09h = Power Unit |
| **12** | Sensor Number | 01h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] = Sensor Specific offset as described in Table 19 |
| **15** | Event Data 2 | Not used |
| **16** | Event Data 3 | Not used |

**Table 19. Power Unit Status Sensor – Sensor Specific Offsets – Next Steps**

| Sensor Specific Offset | | Description | Next Steps |
|---|---|---|---|
| **00h** | Power down | System is powered down. | Informational event |
| **02h** | 240 VA power down | 240 VA power limit was exceeded and the hardware forced a power down. | This could be caused by many things.<br>1. If hardware was recently added, try removing it.<br>2. Remove/replace any add-in adapters.<br>3. Remove/replace the power supply.<br>4. Remove/replace the processors, DIMM, and/or hard drives.<br>5. Remove/replace the boards in the system. |
| **04h** | AC Lost | AC power was removed. | Informational event |
| **05h** | Soft Power Control Failure | Asserted if the system fails to power on due to the following power control sources:<br>• Chassis Control command<br>• PEF action<br>• BMC Watchdog Timer<br>• Power State Retention | This could be caused by the power supply subsystem or system components.<br>1. Verify all power cables and adapters are connected properly (AC cables as well as the cables between the power supply unit (PSU) and system components).<br>2. Cross test the PSU if possible.<br>3. Replace the power subsystem. |

| Sensor Specific Offset | | Description | Next Steps |
|---|---|---|---|
| 06h | Power Unit Failure | Power subsystem experienced a failure. Asserted for one of the following conditions:<br>• Unexpected de-assertion of system POWER_GOOD signal.<br>• System fails to respond to any power control source's attempt to power down the system.<br>• System fails to respond to any hardware power control source's attempt to power on the system.<br>• Power Distribution Board (PDB) failure is detected (applies only to systems that have a PDB). | Indicates a power supply failed.<br>1. Remove and reapply AC power.<br>2. Verify all power cables and adapters are connected properly (AC cables as well as the cables between the PSU and system components).<br>3. Cross test the PSU if possible.<br>4. If the power supply still fails, replace it.<br>5. If the problems still exist, replace the baseboard. |

### 4.4.2 Power Unit Redundancy Sensor

This sensor is enabled on the systems that support redundant power supplies. When a system has AC applied or if it loses redundancy of the power supplies, a message is logged into the SEL.

**Table 20. Power Unit Redundancy Sensors Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| 11 | Sensor Type | 09h = Power Unit |
| 12 | Sensor Number | 02h |
| 13 | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 0Bh (Generic Discrete) |
| 14 | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset as described in Table 21 |
| 15 | Event Data 2 | Not used |
| 16 | Event Data 3 | Not used |

**Table 21. Power Unit Redundancy Sensor – Event Trigger Offset – Next Steps**

| Event Trigger Offset | | Description | Next Steps |
|---|---|---|---|
| 00h | Fully redundant | System is fully operational. | Informational Event |
| 01h | Redundancy lost | System is not running in redundant power supply mode. | This event is accompanied by specific power supply errors (AC lost, PSU failure, and so on). Troubleshoot these events accordingly. |
| 02h | Redundancy degraded | | |
| 03h | Non-redundant, sufficient from redundant | | |
| 04h | Non-redundant, sufficient from insufficient | | |
| 05h | Non-redundant, insufficient | | |
| 06h | Non-redundant, degraded from fully redundant | | |
| 07h | Redundant, degraded from non-redundant | | |

### 4.4.3 Node Auto Shutdown Sensor

The BMC supports a Node Auto Shutdown sensor for logging an SEL event due to an emergency shutdown of a node due to loss of power supply redundancy or PSU CLST throttling due to an over-current warning condition. This sensor is applicable only to multi-node systems.

The sensor is rearmed on power-on (AC or DC power-on transitions).

This sensor is only used for triggering SEL to indicate node or power auto shutdown assertion or deassertion.

**Table 22. Node Auto Shutdown Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| 11 | Sensor Type | 09h = Power Unit |
| 12 | Sensor Number | B8h |
| 13 | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 03h (“digital” discrete) |
| 14 | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset<br>1h = State Asserted |
| 15 | Event Data 2 | Not used |
| 16 | Event Data 3 | Not used |

#### 4.4.3.1 Node Auto Shutdown Sensor – Next Steps

This event is accompanied by specific power supply errors (AC lost, PSU failure, and so on) or other system events. Troubleshoot these events accordingly.

## 4.5 Power Supply

The BMC monitors the power supply subsystem.

### 4.5.1 Power Supply Status Sensors

These sensors report the status of the power supplies in the system. When a system first has AC applied or removed, it can log an event. Also if there is a failure, predictive failure, or a configuration error, it can log an event.

**Table 23. Power Supply Status Sensors Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 08h = Power Supply |
| **12** | Sensor Number | 50h = Power Supply 1 Status<br>51h = Power Supply 2 Status |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| **14** | Event Data 1 | [7:6] – ED2 data in Table 24<br>[5:4] – ED3 data in Table 24<br>[3:0] – Sensor Specific offset as described in Table 24 |
| **15** | Event Data 2 | As described in Table 24 |
| **16** | Event Data 3 | As described in Table 24 |

## Table 24. Power Supply Status Sensor – Sensor Specific Offsets – Next Steps

| Sensor Specific Offset | | Description | ED2 | ED3 | Next Steps |
|---|---|---|---|---|---|
| **00h** | Presence | Power supply detected. | 00b = Unspecified Event Data 2 | 00b = Unspecified Event Data 3 | Informational Event |
| **01h** | Failure | Power supply failed. Check the data in ED2 and ED3 for more details. | 10b = OEM code in Event Data 2<br>• 01h – Output voltage fault<br>• 02h – Output power fault<br>• 03h – Output over-current fault<br>• 04h – Over-temperature fault<br>• 05h – Fan fault | 10b = OEM code in Event Data 3<br>Has the contents of the associated PMBus* Status register. For example, Data 3 will have the contents of the VOLTAGE_STATUS register at the time an Output Voltage fault was detected. Refer to the *PMBus* Specification* for details on specific register contents. | Indicates a power supply failed.<br>1. Remove and reapply AC.<br>2. If the power supply still fails, replace it. |
| **02h** | Predictive Failure | Check the data in ED2 and ED3 for more details. | 10b = OEM code in Event Data 2<br>• 01h – Output voltage warning<br>• 02h – Output power warning<br>• 03h – Output over-current warning<br>• 04h –Over-temperature warning<br>• 05h – Fan warning<br>• 06h – Input under-voltage warning<br>• 07h – Input over-current warning<br>• 08h – Input over-power warning | 10b = OEM code in Event Data 3<br>Has the contents of the associated PMBus* Status register. For example, Data 3 will have the contents of the VOLTAGE_STATUS register at the time an Output Voltage warning was detected. Refer to the *PMBus* Specification* for details on specific register contents. | Depends on the warning event.<br>1. Replace the power supply.<br>2. Verify proper airflow to the system.<br>3. Verify the power source.<br>4. Replace the system boards. |
| **03h** | AC lost | AC removed. | 00b = Unspecified Event Data 2 | 00b = Unspecified Event Data 3 | Informational Event. |
| **06h** | Configuration error | Power supply configuration is not supported. Check the data in ED2 for more details. | 10b = OEM code in Event Data 2<br>• 01h – The BMC cannot access the PMBus* device on the PSU but its FRU device is responding.<br>• 02h – The PMBUS*_REVISION command returns a version number that is not supported (only version 1.1 and 1.2 are supported).<br>• 03h – The PMBus* device does not successfully respond to the PMBUS*_REVISION command.<br>• 04h – The PSU is incompatible with one or more PSUs that are present in the system.<br>• 05h –The PSU FW is operating in a degraded mode (likely due to a failed firmware update). | 00b = Unspecified Event Data 3 | Indicates that at least one of the supplies is not correct for your system configuration.<br>1. Remove the power supply and verify compatibility.<br>2. If the power supply is compatible, it may be faulty. Replace it. |

### 4.5.2 Power Supply Power in Sensors

These sensors log an event when a power supply in the system is exceeding its AC power in threshold.

**Table 25. Power Supply Power In Sensors Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 0Bh = Other Units |
| **12** | Sensor Number | 54h = Power Supply 1 Status<br>55h = Power Supply 2 Status |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 01h (Threshold) |
| **14** | Event Data 1 | [7:6] – 01b = Trigger reading in Event Data 2<br>[5:4] – 01b = Trigger threshold in Event Data 3<br>[3:0] – Event Trigger Offset as described in Table 26 |
| **15** | Event Data 2 | Reading that triggered event |
| **16** | Event Data 3 | Threshold value that triggered event |

The following table describes the severity of each of the event triggers for both assertion and deassertion.

**Table 26. Power Supply Power In Sensor – Event Trigger Offset – Next Steps**

| Event Trigger Offset | | Assertion Severity | Deassertion Severity | Description | Next Steps |
|---|---|---|---|---|---|
| **07h** | Upper non-critical going high | Degraded | OK | PMBus* feature to monitor power supply power consumption. | If you see this event, the system is pulling too much power on the input for the PSU rating.<br>• Verify the power budget is within the specified range.<br>• Check http://www.intel.com/p/en_US/support/ for the power budget tool for your system. |
| **09h** | Upper critical going high | Non-fatal | Degraded | | |

### 4.5.3 Power Supply Current Out % Sensors

PMBus*-compliant power supplies may monitor the current output of the main 12 V voltage rail and report the current usage as a percentage of the maximum power output for that rail.

**Table 27. Power Supply Current Out % Sensors Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 03h = Current |
| **12** | Sensor Number | 58h = Power Supply 1 Current Out %<br>59h = Power Supply 2 Current Out % |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 01h (Threshold) |
| **14** | Event Data 1 | [7:6] – 01b = Trigger reading in Event Data 2<br>[5:4] – 01b = Trigger threshold in Event Data 3<br>[3:0] – Event Trigger Offset as described in Table 28 |
| **15** | Event Data 2 | Reading that triggered event |
| **16** | Event Data 3 | Threshold value that triggered event |

The following table describes the severity of each of the event triggers for both assertion and deassertion.

**Table 28: Power Supply Current Out % Sensor – Event Trigger Offset – Next Steps**

| Event Trigger Offset | | Assertion Severity | Deassertion Severity | Description | Next Steps |
|---|---|---|---|---|---|
| 07h | Upper non-critical going high | Degraded | OK | PMBus* feature to monitor power supply power consumption. | If you see this event, the system is using too much power on the output for the PSU rating.<br>1. Verify the power budget is within the specified range.<br>• Check http://www.intel.com/p/en_US/support/ for the power budget tool for your system. |
| 09h | Upper critical going high | Non-fatal | Degraded | | |

### 4.5.4 Power Supply Temperature Sensors

The BMC monitors one or two power supply temperature sensors for each installed PMBus*-compliant power supply.

**Table 29. Power Supply Temperature Sensors Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| 11 | Sensor Type | 01h = Temperature |
| 12 | Sensor Number | 5Ch = Power Supply 1 Temperature<br>5Dh = Power Supply 2 Temperature |
| 13 | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 01h (Threshold) |
| 14 | Event Data 1 | [7:6] – 01b = Trigger reading in Event Data 2<br>[5:4] – 01b = Trigger threshold in Event Data 3<br>[3:0] – Event Trigger Offset as described in Table 30 |
| 15 | Event Data 2 | Reading that triggered event |
| 16 | Event Data 3 | Threshold value that triggered event |

The following table describes the severity of each of the event triggers for both assertion and deassertion.

**Table 30. Power Supply Temperature Sensor – Event Trigger Offset – Next Steps**

| Event Trigger Offset | | Assertion Severity | Deassertion Severity | Description | Next Steps |
|---|---|---|---|---|---|
| 07h | Upper non-critical going high | Degraded | OK | An upper non-critical or critical temperature threshold has been crossed. | Check for clear and unobstructed airflow into and out of the chassis.<br>1. Ensure the SDR is programmed and correct chassis has been selected.<br>2. Ensure there are no fan failures.<br>3. Ensure the air used to cool the system is within the thermal specifications for the system (typically below 35°C). |
| 09h | Upper critical going high | Non-fatal | Degraded | | |

### 4.5.5 Power Supply Fan Tachometer Sensors

The BMC polls each installed power supply using the PMBus* fan status commands to check for failure conditions for the power supply fans.

**Table 31. Power Supply Fan Tachometer Sensors Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 04h = Fan |
| **12** | Sensor Number | A0h = Power Supply 1 Fan Fail[3] (PS1 Fan1 Fail)<br>A1h =Power Supply 1 Fan 2 Fail[3] (PS1 Fan2 Fail)<br>A4h = Power Supply 2 Fan Fail[3] (PS2 Fan1 Fail)<br>A5h = Power Supply 2 Fan 2 Fail[3] (PS2 Fan2 Fail) |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 03h (“digital” Discrete) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset = 1h = State Asserted |
| **15** | Event Data 2 | Not used |
| **16** | Event Data 3 | Not used |

#### 4.5.5.1 Power Supply Fan Tachometer Sensors – Next Steps

These events only get generated in systems with PMBus*-capable power supplies and, normally, when the airflow is obstructed to the power supply:

1. Remove and then reinstall the power supply to see whether something might have temporarily caused the fan failure.
2. Swap the power supply with another one to see whether the problem stays with the location or follows the power supply.
3. Replace the power supply depending on the outcome of steps 1 and 2.
4. Ensure the latest FRUSDR update has been run and the correct chassis is detected or selected.

# 5. Cooling Subsystem

## 5.1 Fan Sensors

There are three types of fan sensors that can be present on Intel® Server Systems: speed, presence, and redundancy. The last two are only present in systems with hot-swap redundant fans.

### 5.1.1 Fan Tachometer Sensors

Fan tachometer sensors monitor the rpm signal on the relevant fan headers on the platform. Fan speed sensors are threshold-based sensors. Usually, they only have lower (critical) thresholds set, so that an SEL entry is only generated if the fan spins too slowly.

**Table 32. Fan Tachometer Sensors Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | `04h` = Fan |
| **12** | Sensor Number | `30h-3Fh` (Chassis specific) |
| **13** | Event Direction and Event Type | [7] Event direction<br>`0b` = Assertion Event<br>`1b` = Deassertion Event<br>[6:0] Event Type = `01h` (Threshold) |
| **14** | Event Data 1 | [7:6] – `01b` = Trigger reading in Event Data 2<br>[5:4] – `01b` = Trigger threshold in Event Data 3<br>[3:0] – Event Trigger Offset as described in Table 33 |
| **15** | Event Data 2 | Reading that triggered event |
| **16** | Event Data 3 | Threshold value that triggered event |

The following table describes the severity of each of the event triggers for both assertion and deassertion.

**Table 33. Fan Tachometer Sensor – Event Trigger Offset – Next Steps**

| Event Trigger Offset | | Assertion Severity | Deassertion Severity | Description | Next Steps |
|---|---|---|---|---|---|
| **00h** | Lower non-critical going low | Degraded | OK | The fan speed has dropped below its lower non-critical threshold. | A fan speed error on a new system build is typically not caused by the fan spinning too slowly; instead it is caused by the fan being connected to the wrong header (the BMC expects them on certain headers for each chassis and logs this event if there is no fan on that header).<br>1. Refer to the Quick Start Guide or the Service Guide to identify the correct fan headers to use.<br>2. Ensure the latest FRUSDR update has been run and the correct chassis is detected or selected.<br>3. If this was done already, the event may be a sign of impending fan failure (although this only normally applies if the system has been in use for a while). Replace the fan. |
| **02h** | Lower critical going low | non-fatal | Degraded | The fan speed has dropped below its lower critical threshold. | |

### 5.1.2 Fan Presence and Redundancy Sensors

Fan presence sensors are only implemented for hot-swap fans, and require an additional pin on the fan header. Fan redundancy is an aggregate of the fan presence sensors and warn when redundancy is lost. Typically, the redundancy mode on Intel Server Systems is an n+1 redundancy. Meaning, if one fan fails, there are still sufficient fans to cool the system, but it is no longer redundant. Although, other modes are also possible.

**Table 34. Fan Presence Sensors Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 04h = Fan |
| **12** | Sensor Number | 40h-45h (Chassis specific) |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 08h (Generic “digital” Discrete) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset as described in Table 35 |
| **15** | Event Data 2 | Not used |
| **16** | Event Data 3 | Not used |

The following table describes the severity of each of the event triggers for both assertion and deassertion.

**Table 35. Fan Presence Sensors – Event Trigger Offset – Next Steps**

| Event Trigger Offset | | Assertion Severity | Deassertion Severity | Description | Next Steps |
|---|---|---|---|---|---|
| **01h** | Device Present | OK | Degraded | Assertion – A fan was inserted. This event may also get logged when the BMC initializes when AC is applied. | Informational only. |
| | | | | Deassertion – A fan was removed, or was not present at the expected location when the BMC initialized. | These events only get generated in the systems with hot-swappable fans, and normally only when a fan is physically inserted or removed. If fans were not physically removed:<br>1. Use the Quick Start Guide to check whether the right fan headers were used.<br>2. Swap the fans round to see whether the problem stays with the location or follows the fan.<br>3. Replace the fan or fan wiring/housing depending on the outcome of step 2.<br>4. Ensure the latest FRUSDR update has been run and the correct chassis is detected or selected. |

### Table 36. Fan Redundancy Sensors Typical Characteristics

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 04h = Fan |
| **12** | Sensor Number | 0Ch |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 0Bh (Generic Discrete) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset as described in Table 37 |
| **15** | Event Data 2 | Not used |
| **16** | Event Data 3 | Not used |

The following table describes the severity of each of the event triggers for both assertion and deassertion.

### Table 37. Fan Redundancy Sensor – Event Trigger Offset – Next Steps

| Event Trigger Offset | | Description | Next Steps |
|---|---|---|---|
| **00h** | Fully redundant | The system has lost one or more fans and is running in non-redundant mode. There are enough fans to keep the system properly cooled, but fan speeds will boost. | Fan redundancy loss indicates failure of one or more fans. Look for lower (non-) critical fan errors, or fan removal errors in the SEL, to indicate which fan is causing the problem, and follow the troubleshooting steps for these event types. |
| **01h** | Redundancy lost | | |
| **02h** | Redundancy degraded | | |
| **03h** | Non-redundant, sufficient from redundant | | |
| **04h** | Non-redundant, sufficient from insufficient | | |
| **05h** | Non-redundant, insufficient | The system has lost fans and may no longer be able to cool itself adequately. Overheating may occur if this situation remains for a longer period of time. | |
| **06h** | Non-redundant, degraded from fully redundant | The system has lost one or more fans and is running in non-redundant mode. There are enough fans to keep the system properly cooled, but fan speeds will boost. | |
| **07h** | Redundant, degraded from non-redundant | The system has lost one or more fans and is running in a degraded mode, but still is redundant. There are enough fans to keep the system properly cooled. | |

## 5.2 Temperature Sensors

There are a variety of temperature sensors that can be implemented on Intel® Server Systems. They are split into various types each with their own events that can be logged.

- Threshold-based Temperature
- Thermal Margin
- Processor Thermal Control %
- Processor DTS Thermal Margin (Monitor only)
- Discrete Thermal
- DIMM Thermal Trip

### 5.2.1 Threshold-based Temperature Sensors

Threshold-based temperature sensors are sensors that report an actual temperature. These are linear, threshold-based sensors. In most Intel Server Systems, multiple sensors are defined: front panel temperature and baseboard temperature. There are also multiple other sensors that can be defined and are platform-specific. Most of these sensors typically have upper and lower thresholds set – upper to warn in case of an over-temperature situation and lower to warn against sensor failure (temperature sensors typically read out 0 if they stop working).

**Table 38. Temperature Sensors Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 01h = Temperature |
| **12** | Sensor Number | See Table 40 |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 01h (Threshold) |
| **14** | Event Data 1 | [7:6] – 01b = Trigger reading in Event Data 2<br>[5:4] – 01b = Trigger threshold in Event Data 3<br>[3:0] – Event Trigger Offset as described in Table 39 |
| **15** | Event Data 2 | Reading that triggered event |
| **16** | Event Data 3 | Threshold value that triggered event |

**Table 39. Temperature Sensors Event Triggers**

| Event Trigger | | Assertion Severity | Deassertion Severity | Description |
|---|---|---|---|---|
| **00h** | Lower non-critical going low | Degraded | OK | The temperature has dropped below its lower non-critical threshold. |
| **02h** | Lower critical going low | Non-fatal | Degraded | The temperature has dropped below its lower critical threshold. |
| **07h** | Upper non-critical going high | Degraded | OK | The temperature has gone over its upper non-critical threshold. |
| **09h** | Upper critical going high | Non-fatal | Degraded | The temperature has gone over its upper critical threshold. |

## Table 40. Temperature Sensors – Next Steps

| Sensor # | Sensor Name | Next Steps |
|---|---|---|
| 21h (M50CYP) | Front Panel Temp | If the front panel temperature reads zero, check:<br>• It is connected properly.<br>• The SDR has been programmed correctly for your chassis.<br>If the front panel temperature is too high:<br>• Check the cooling of your server room. |
| 14h (M50CYP) | Baseboard Temperature 5 (BB Lft Rear Temp) | Check for clear and unobstructed airflow into and out of the chassis.<br>• Ensure the SDR is programmed and correct chassis has been selected.<br>• Ensure there are no fan failures.<br>• Ensure the air used to cool the system is within the thermal specifications for the system (typically below 35°C). |
| 14h (D50TNP) | Baseboard Temperature 5 (BB Outlet Temp) | |
| 20h (M50CYP) | Baseboard Temperature 1 (BB P0 VR Temp) | |
| 21h (D50TNP) | Baseboard Temperature 2 (BB inlet Temp) | |
| 22h | PCH Temperature(PCH Temp) | |
| 23h (M50CYP) | Baseboard Temperature 2 (BB P1 VR Temp)) | |
| 24h (M50CYP) | Baseboard Temperature 3 (BB BMC Temp) | |
| 24h (D50TNP) | Baseboard Temperature3 (CPU0 VR Temp) | |
| 25h (M50CYP) | Baseboard Temperature 4 (BB M.2 Temp) | |
| 25h (D50TNP) | Network Interface Controller Temperature (LAN Temp) | |
| 26h (M50CYP) | OCP Module Temperature (OCP Mod Temp) | |
| 26h (D50TNP) | Baseboard Temperature (CPU1 VR Temp) | |
| 27h (D50TNP) | PCI Riser 1 Temperature (Riser 1 Temp) | |
| 28h (D50TNP) | U.2 NVMe* Temperature (U.2 SSD Temp1) | |
| 29h (D50TNP) | M.2 NVMe* Temperature (M.2 Temp1) | |
| 2Ah (D50TNP) | PDB Temperature (PDB Temperature) | |
| 2Ch (M50CYP) | PCI Riser 1 Temperature (Riser 1 Temp) | |
| 2Ch (D50TNP) | PCI Riser 2 Temperature (Riser 2 Temp) | |
| 2Dh (M50CYP) | PCI Riser 2 Temperature (Riser 2 Temp) | |
| 2Dh (D50TNP) | U.2 NVMe* Temperature (U.2 SSD Temp2) | |
| 2Eh | Exit Air Temperature (Exit Air Temp) | |

| Sensor # | Sensor Name | Next Steps |
|---|---|---|
| 2Fh (M50CYP) | PCI Riser 3 Temperature (Riser 3 Temp) | |
| 2Fh (D50TNP) | Network Interface Controller Temperature (LAN NIC Temp) | |
| 44h (D50TNP) | EV CPU 1 VR Temperature (EV CPU1VR Temp) | |
| 45h (D50TNP) | Memory 0 VRD Temperature (Mem 0 VR Temp) | |
| 46h (D50TNP) | Memory 1 VRD Temperature (Mem 1 VRD Temp) | |
| 47h (D50TNP) | M.2 NVMe* Temperature (M.2 Temp 2) | |
| 4Ch (M50CYP) | Processor 1 VR controller temp (P0 VR Ctrl Temp) | |
| 4Dh (M50CYP) | Processor 2 VR controller temp (P1 VR Ctrl Temp) | |
| 60h (D50TNP) | EDSFF Dock Temperature (Dock Temp) | |
| 61h (D50TNP) | Max RSSD temperature (RSSD Aggr Thrm 1) | |
| 62h (D50TNP) | Max RSSD temperature (RSSD Aggr Thrm 2) | |
| 69h (M50CYP) | Midplane 1 Temp (Left Swch Temp) | |
| 6Ah (M50CYP) | Midplane 2 Temp (Right Swch Temp) | |
| 6Bh (M50CYP) | SAS Module Temperature (SAS Mod Temp) | |
| 85h (M50CYP) | Processor 1 DTS Temperature (P0 D1 Abs Temp) | |
| 85h (D50TNP) | Processor 1 DTS Temperature (P1 Temperature) | |
| 86h (D50TNP) | Processor 2 DTS Temperature (P2 Temperature) | |
| 88h (D50TNP) | GPGPU Riser Temperature (GPGPU Riser) | |
| 8Ah (M50CYP) | Processor 2 DTS Temperature (P1 D1 Abs Temp) | |
| 93h (D50TNP) | CPU0 VCCANA Temp | |
| 94h (D50TNP) | CPU1 VCCANA Temp | |

### 5.2.2 Thermal Margin Sensors

Margin sensors are also linear sensors but typically report a negative value. This is not an actual temperature, but in fact an offset to a critical temperature. Values reported are seen as number of degrees below a critical temperature for the particular component.

The BMC supports DIMM aggregate temperature margin IPMI sensors. The temperature readings from the physical temperature sensors on each DIMM, such as temperature sensor on DIMM (TSOD), are aggregated into IPMI temperature margin sensors for groupings of DIMM slots, the partitioning of which is platform/SKU specific and generally corresponding to fan domains.

The BMC supports global aggregate temperature margin IPMI sensors. There may be as many unique global aggregate sensors as there are fan domains. Each sensor aggregates the readings of multiple other IPMI temperature sensors supported by the BMC firmware. The mapping of child-sensors into each global aggregate sensor is SDR-configurable. The primary usage for these sensors is to trigger turning off fans when a lower threshold is reached.

**Table 41. Thermal Margin Sensors Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 01h = Temperature |
| **12** | Sensor Number | See Table 43 |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 01h (Threshold) |
| **14** | Event Data 1 | [7:6] – 01b = Trigger reading in Event Data 2<br>[5:4] – 01b = Trigger threshold in Event Data 3<br>[3:0] – Event Triggers as described in Table 42 |
| **15** | Event Data 2 | Reading that triggered event |
| **16** | Event Data 3 | Threshold value that triggered event |

**Table 42. Thermal Margin Sensors Event Triggers**

| Event Trigger | | Assertion Severity | Deassertion Severity | Description |
|---|---|---|---|---|
| **07h** | Upper non-critical going high | Degraded | OK | The thermal margin has gone over its upper non-critical threshold. |
| **09h** | Upper critical going high | non-fatal | Degraded | The thermal margin has gone over its upper critical threshold. |

**Table 43. Thermal Margin Sensors – Next Steps**

| Sensor # | Sensor Name | Next Steps |
|---|---|---|
| **74h** | P1 Therm Margin | Not a logged SEL event. Sensor is used for thermal management of the processor. |
| **75h** | P2 Therm Margin | |
| **76h** | P3 Therm Margin | |
| **77h** | P4 Therm Margin | |
| **5Eh** | PCIe* Aggregate Thermal Margin (PCIe* Aggr Margin) | 1. Check for clear and unobstructed airflow into and out of the chassis.<br>2. Ensure the SDR is programmed and correct chassis has been selected.<br>3. Ensure there are no fan failures. |
| **B0h** | P1 DIMM Thrm Mrgn1 | |

| Sensor # | Sensor Name | Next Steps |
|---|---|---|
| B1h | P1 DIMM Thrm Mrgn2 | Ensure the air used to cool the system is within the thermal specifications for the system (typically below 35°C). |
| B2h | P2 DIMM Thrm Mrgn1 | |
| B3h | P2 DIMM Thrm Mrgn2 | |
| C8h | Global Aggregate Temperature Margin 1 (Agg Therm Mgn 1) | |
| C9h | Global Aggregate Temperature Margin 2 (Agg Therm Mgn 2) | |

### 5.2.3 Processor Thermal Control Sensors

The BMC firmware monitors the percentage of time that a processor has been operationally constrained over a given time window (nominally six seconds) due to internal thermal management algorithms engaging to reduce the temperature of the device. This monitoring is instantiated as one IPMI analog/threshold sensor per processor package.

If this is not addressed, the processor will overheat and shut down the system to protect itself from damage.

**Table 44. Processor Thermal Control Sensors Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| 11 | Sensor Type | 01h = Temperature |
| 12 | Sensor Number | 78h = Processor 1 Thermal Control % (P1 Therm Ctrl %)<br>79h = Processor 2 Thermal Control % (P2 Therm Ctrl %) |
| 13 | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 01h (Threshold) |
| 14 | Event Data 1 | [7:6] – 01b = Trigger reading in Event Data 2<br>[5:4] – 01b = Trigger threshold in Event Data 3<br>[3:0] – Event Triggers as described in Table 45 |
| 15 | Event Data 2 | Reading that triggered event |
| 16 | Event Data 3 | Threshold value that triggered event |

**Table 45. Processor Thermal Control Sensors Event Triggers**

| Event Trigger | | Assertion Severity | Deassertion Severity | Description |
|---|---|---|---|---|
| 07h | Upper non-critical going high | Degraded | OK | The thermal margin has gone over its upper non-critical threshold. |
| 09h | Upper critical going high | Non-fatal | Degraded | The thermal margin has gone over its upper critical threshold. |

#### 5.2.3.1 Processor Thermal Control % Sensors – Next Steps

These events normally occur due to failures of the thermal solution:

1. Verify heat sink is properly attached and has thermal grease.
2. If the system has a heat sink fan, ensure the fan is spinning.
3. Check all system fans are operating properly.
4. Check that the air used to cool the system is within limits (typically 35 °C).

### 5.2.4 Processor DTS Thermal Margin Sensors

The Intel® Xeon® Scalable processor family incorporates a DTS-based thermal specification conformance capability. This capability allows a much more accurate control of the thermal solution and will enable lower fan speeds and lower fan power consumption. The BMC accesses the margin reading via PECI commands to the processor.

The main usage of this sensor is as an input to the BMC's fan control algorithms. Thresholds are not set and alert generation is not enabled for these sensors.

**Table 46. Processor DTS Thermal Margin Sensors Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 01h = Temperature |
| **12** | Sensor Number | M50TNP:<br>83h = Processor 1 DTS Thermal Margin<br>88h = Processor 2 DTS Thermal Margin<br>D50TNP:<br>83h = Processor 1 DTS Thermal Margin<br>84h = Processor 2 DTS Thermal Margin |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 01h (Threshold) |
| **14** | Event Data 1 | [7:6] – 01b = Trigger reading in Event Data 2<br>[5:4] – 01b = Trigger threshold in Event Data 3<br>[3:0] – Event Triggers as described in Table 13 |
| **15** | Event Data 2 | Reading that triggered event, FFh or not present if unspecified. |
| **16** | Event Data 3 | Threshold value that triggered event, FFh or not present if unspecified. If present, byte 2 must be present. |

### 5.2.5 Aggregate VRD-Hot Sensor

The BMC monitors processor and memory VRD_HOT_N signals. The VRD_HOT_N signals are routed to the associated processor PROCHOT_N or MEMHOT#_N input in order to initiate throttling to reduce the power draw, thereby indirectly lowering the VRD temperature.

There is one processor and two memory VRD_HOT# signals per CPU slot. The BMC manifests one discrete IPMI sensor that aggregates the status of all of the VRD_HOT_N signals in the system. The associated SEL record includes an OEM bit-map indicating the state of each VRD_HOT_N signal when the record was logged. The sensor will only go into a de-asserted state when all signals have de-asserted.

**Table 47. VRD-Hot Sensors Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 01h = Temperature |
| **12** | Sensor Number | 90h = VRD-Hot |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 05h (Digital Discrete) |
| **14** | Event Data 1 | [7:6] - 10b = OEM code in Byte 2<br>[5:4] - 10b = OEM code in Byte 3<br>[3:0] - Offset from Event/Reading code |
| **15** | Event Data 2 | [7:0] - CPU VRHOT Map<br>[Bit 0] = CPU 0<br>[Bit 1] = CPU 1 |

| Byte | Field | Description |
|---|---|---|
| 16 | Event Data 3 | [7:0] - Memory VRHOT Map<br>[Bit 0] = CPU 0 Memory A/B/C<br>[Bit 1] = CPU 0 Memory D/E/F<br>[Bit 2] = CPU 1 Memory A/B/C<br>[Bit 3] = CPU 1 Memory D/E/F |

### 5.2.6 Discrete Thermal Sensors

Discrete thermal sensors do not report a temperature at all; instead they report an overheating event of some kind. For example, VRD Hot (voltage regulator is overheating) or processor Thermal Trip (the processor got so hot that its over-temperature protection was triggered and the system was shut down to prevent damage).

#### 5.2.6.1 PCH Thermal Trip Sensor

**Table 48. PCH Thermal Trip Sensors Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| 11 | Sensor Type | 19h = Chip Set |
| 12 | Sensor Number | 0Dh = PCH Thermal Trip |
| 13 | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| 14 | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset = 01h (State Asserted, South Side Bridge (SSB) overheated) |
| 15 | Event Data 2 | Not used |
| 16 | Event Data 3 | Not used |

#### 5.2.6.2 VRD HOT Sensor

**Table 49. VRD HOT Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| 11 | Sensor Type | 01h = Temperature |
| 12 | Sensor Number | 90h = SSB Thermal Trip |
| 13 | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 05h (Digital Discrete) |
| 14 | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset = 01h (Limit Exceeded) |
| 15 | Event Data 2 | Processor VRD HOT bitmap<br>[7:4] - Reserved<br>[3] - CPU4<br>[2] - CPU3<br>[1] - CPU2<br>[0] – CPU1 |
| 16 | Event Data 3 | Memory VRD HOT bitmap<br>[7] – CPU4, DIMM Channel3/4<br>[6] – CPU4, DIMM Channel1/2 |

| Byte | Field | Description |
|---|---|---|
| | | [5] – CPU3, DIMM Channel3/4<br>[4] – CPU3, DIMM Channel1/2<br>[3] – CPU2, DIMM Channel3/4<br>[2] – CPU2, DIMM Channel1/2<br>[1] – CPU1, DIMM Channel3/4<br>[0] – CPU1, DIMM Channel1/2 |

#### 5.2.6.3 Discrete Thermal Sensors – Next Steps

1. Check for clear and unobstructed airflow into and out of the chassis.
2. Ensure the SDR is programmed and correct chassis has been selected.
3. Ensure there are no fan failures.
4. Ensure the air used for cooling the system is within the thermal specifications for the system (typically below 35 °C).

### 5.2.7 Liquid Cooling

The liquid cooling solution falls into two categories: the Liquid Assisted Air Cooling (LAAC) and Liquid Cooling Rack(LCR). The LAAC consists of components such as a cold plate, pump, Heat Exchanger (HEX), liquid tube, coolant etc. The LCR consists of components such as a cold plate, pump, liquid tubes, coolant, rack CDU, even facility cooling devices, etc.

For the BMC to support liquid cooling, it must provide the following functionalities:

- Pump tachometer sensor
- System shutdown when pump reading crosses critical high or critical low

For Intel® Server Systems supporting the Intel® Xeon® processor Scalable family to support a liquid cooling feature, a special FRU is needed.

#### 5.2.7.1 Pump Tachometer Sensors

The pump tachometer sensor reading is proportional to the pump speed. This monitoring capability is instantiated in the form of IPMI analog/threshold sensors. Sensor reading is available only during DC-on.

The pump sensor is a manual re-arm sensor. Once the sensor status crosses critical high or critical low, the sensor status can't auto-recover because it may be damaged. The BMC will shut the system down in this situation and the user should check and replace the sensor if it is damaged. The BMC will re-initialize the sensor at the next system power-on after the problem has been handled.

#### 5.2.7.2 IPMI Sensor Characteristics

- Event reading type code: 01h (Threshold)
- Sensor type code: 0Ah (Cooling Device)
- Re-arm type: Manual
- Configured thresholds: Lower/Upper critical/non-critical
- Event generation: Assertion/de-assertion events for all supported thresholds

**Table 50. Threshold-Based Voltage Sensors Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | `0Ah` = Cooling Device |
| **12** | Sensor Number | `3Ch-3Dh`= Pump Tachometer Sensors |
| **13** | Event Direction and Event Type | [7] Event direction<br>`0b` = Assertion Event |

| Byte | Field | Description |
|---|---|---|
| | | 1b = Deassertion Event<br>[6:0] Event Type = 01h (Threshold) |
| 14 | Event Data 1 | [7:6] – 01b = Trigger reading in Event Data 2<br>[5:4] – 01b = Trigger threshold in Event Data 3<br>[3:0] – Event Triggers as described in Table 13 |
| 15 | Event Data 2 | Reading that triggered event, FFh or not present if unspecified. |
| 16 | Event Data 3 | Threshold value that triggered event, FFh or not present if unspecified. If present, byte 2 must be present. |

The following table describes the severity of each of the event triggers for both assertion and deassertion.

**Table 51. Threshold-Based Voltage Sensors Event Triggers**

| Event Trigger | | Assertion Severity | Deassertion Severity | Description |
|---|---|---|---|---|
| 00h | Lower non-critical going low | Degraded | OK | The voltage has dropped below its lower non-critical threshold. |
| 02h | Lower critical going low | Non-fatal | Degraded | The voltage has dropped below its lower critical threshold. |
| 07h | Upper non-critical going high | Degraded | OK | The voltage has gone over its upper non-critical threshold. |
| 09h | Upper critical going high | Non-fatal | Degraded | The voltage has gone over its upper critical threshold. |

## 5.2.8 DIMM Thermal Trip Sensors

The BMC supports DIMM Thermal Trip monitoring that is instantiated as one aggregate IPMI discrete sensor per processor. When a DIMM Thermal Trip occurs, the system hardware automatically powers down the server and the BMC asserts the sensor offset and logs an event.

**Table 52. DIMM Thermal Trip Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| 11 | Sensor Type | 0Ch = Memory |
| 12 | Sensor Number | C0h = Processor 1 DIMM Thermal Trip<br>C1h = Processor 2 DIMM Thermal Trip |
| 13 | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| 14 | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset = 0A = Critical over temperature |
| 15 | Event Data 2 | Not used |
| 16 | Event Data 3 | [7:5] – Socket ID<br>0–3 = CPU1-4<br>[4:3] – Channel<br>0–3 = Channel A, B, C, D for CPU1<br>Channel E, F, G, H for CPU2<br>Channel J, K, L, M for CPU3<br>Channel N, P, R, T for CPU4<br>[2:0] – DIMM<br>0–2 = DIMM 1–3 on Channel |

#### 5.2.8.1 DIMM Thermal Trip Sensors – Next Steps

1. Check for clear and unobstructed airflow into and out of the chassis.
2. Ensure the SDR is programmed and correct chassis has been selected.
3. Ensure there are no fan failures.
4. Ensure the air used to cool the system is within the thermal specifications for the system (typically below 35 °C).

### 5.2.9 NVMe* Thermal Status

The BMC supports NVMe* temperature and critical warning monitoring that is instantiated as one aggregate IPMI discrete sensor per HSBP. Each backplane provides a single temperature sensor. The aggregate sensor reports the temperature of the hottest drive inserted into the HSBP. The BMC uses the NVMe* drive's SMBus* Management Interface ($I^2C$ address 0xD4 byte 3) to retrieve temperature data. Because discrete temperature sensors were deprecated early in the NVMe* drive life-cycle, drive temperature is never acquired from a discrete sensor on the drive. The BMC, as a result, can only rely on temperature data acquired via the Management Interface.

#### 5.2.9.1 NVMe* Temperature Sensor

The NVMe* temperature SEL occurs when the aggregate temperature sensor reading reaches the upper critical threshold. In order to report which of several drives are over temp, OEM Extended data bytes are used. The remainder of the SEL uses standard temperature sensor decoding rules. The NVMe* Temperature SEL uses a new format for the OEM Extended data bytes.

**Table 53. NVMe* Temperature Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 01h = Temperature |
| **12** | Sensor Number | 91h = NVMe 1 Therm Mgn<br>92h = NVMe 2 Therm Mgn<br>93h = NVMe 3 Therm Mgn |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 01h (Threshold) |
| **14** | Event Data 1 | [7:6] – 10b = Trigger reading in Event Data 2<br>[5:4] – 10b = Trigger threshold in Event Data 3<br>[3:0] – 0001b = OEM Event Trigger in Table 39 |
| **15** | Event Data 2 | Reading that triggered event |
| **16** | Event Data 3 | Threshold value that triggered event |
| **17** | Extended Data 1 | [7] = Extended format indicator<br>[6:4] = # Extended Data byte<br>[3:2] = reserved<br>[1:0] = Existing severity encoding |
| **18** | Extended Data 2 | 0 = Reserved<br>1 = Drive Position Format<br>2 = Critical Warning Format<br>3 = Add in card format<br>4-0xff = Reserved |
| **19** | Extended Data 3 | For Format = 1 (Drive Position Format), ExtData[2] is a bitfield indicating which drives are over temperature. Bit 0 set indicates Drive 0 is over temp. Bit 7 set indicates Drive 7 is over temp. |
| **20** | Extended Data 4 | If Extended Data 1 indicate this byte is included, same format with above byte. |
| **21** | Extended Data 5 | If Extended Data 1 indicate this byte is included, same format with above byte. |
| **22** | Extended Data 6 | If Extended Data 1 indicate this byte is included, same format with above byte. |

| Byte | Field | Description |
|---|---|---|
| **23** | Extended Data 7 | If Extended Data 1 indicate this byte is included, same format with above byte. |
| **24** | Extended Data 8 | If Extended Data 1 indicate this byte is included, same format with above byte. |

#### 5.2.9.2 NVMe* Critical Warning Sensor

The critical warning sensor aggregates a set of SMART drive warning flags. For the first warning bit asserted for a drive, a corresponding critical warning SEL is generated.

**Table 54. NVMe* Critical Warning Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 0Dh=Disk drive |
| **12** | Sensor Number | 94h = NVMe* 1 Crit Warn<br>95h = NVMe* 2 Crit Warn<br>96h = NVMe* 3 Crit Warn |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 73h (OEM Discrete) |
| **14** | Event Data 1 | 0x8x (low nibble is the drive causing the event, zero based count) |
| **15** | Event Data 2 | SMART warning byte defined in Table 55 for the drive number indicated by the low nibble of ED1. |
| **16** | Event Data 3 | Not used |

**Table 55. SMART Warning Bits**

| Bit | Assertion value | Message |
|---|---|---|
| **0** | 1 | Spare space below threshold |
| **1** | 1 | Temperature above or below threshold |
| **2** | 1 | NVM reliability degraded |
| **3** | 1 | In read-only mode |
| **4** | 1 | Volatile backup service failed |
| **5–7** | X | Reserved |

## 5.3 System Airflow Monitoring Sensor

The BMC provides an IPMI sensor to report the volumetric system airflow in cubic feet per minute (CFM). The airflow in CFM is calculated based on the system fan pulse width modulation (PWM) values. The specific PWM used to determine the CFM is SDR-configurable. The relationship between PWM and CFM is based on a lookup table in an OEM SDR.

The airflow data is used in the calculation for exit air temperature monitoring. It is exposed as an IPMI sensor to allow a data center management application to access this data for use in rack-level thermal management.

This sensor is informational only and does not log events into the SEL.

# 6. Processor Subsystem

Intel® Server Systems report multiple processor-centric sensors in the SEL.

## 6.1 Processor Status Sensor

The BMC provides an IPMI sensor of type processor for monitoring status information for each processor slot. If an event state (sensor offset) has been asserted, it remains asserted until one of the following happens:

- A rearm Sensor Events command is executed for the processor status sensor.
- AC or DC power cycle, system reset, or system boot occurs.

CPU Presence status is not saved across AC power cycles and therefore does not generate a deassertion after cycling AC power.

**Table 56. Process Status Sensors Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 07h = Processor |
| **12** | Sensor Number | 70h = Processor 1 Status<br>71h = Processor 2 Status |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| **14** | Event Data 1 | [7:6] 10b = OEM code in Byte 2<br>[5:4] 00b = Unspecified Byte 3<br>[3:0] – Event Trigger Offset as described in Table 57 |
| **15** | Event Data 2 | IERR Type<br>[7:5] – Reserved<br>[4:0] – Type<br>00000b = Unspecified<br>00001b = CPU/VR mismatch<br>00010b = Non-boot core FIVR fault<br>00011b = Non-boot uncore FIVR fault<br>Other values reserved<br>Processor ThermTrip cause<br>[7:2] – Reserved<br>[1:0] – Type<br>00b = Processor non-recoverable ThermTrip temperature reached.<br>01b = boot FIVR fault<br>Other values reserved<br>Configuration Error<br>FFh |
| **16** | Event Data 3 | FFh |

**Table 57. Processor Status Sensors – Next Steps**

| Event Trigger Offset | Processor Status | Next Steps |
|---|---|---|
| 0h | Internal error (IERR) | 1. Cross test the processors.<br>2. Replace the processors depending on the results of the test. |
| 1h | Thermal trip/FIVR | This event normally only happens due to failures of the thermal solution:<br>1. Verify heat sink is properly attached and has thermal grease.<br>2. If the system has a heat sink fan, ensure the fan is spinning.<br>3. Check all system fans are operating properly.<br>4. Check that the air used to cool the system is within limits (typically 35 °C). |
| 5h | Configuration error | 1. Cross test the processors.<br>2. Replace the processors depending on the results of the test. |
| 7h | Processor presence detected | Informational Event |

## 6.2 Internal Error Sensor

When the Catastrophic Error signal (CATERR#) stays asserted, it is a sign that something serious has gone wrong in the hardware. The BMC monitors this signal and reports when it stays asserted.

**Table 58. Internal Error Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| 11 | Sensor Type | 07h = Processor |
| 12 | Sensor Number | 80h |
| 13 | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 03h (Digital Discrete) |
| 14 | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset = 1h (State Asserted) |
| 15 | Event Data 2 | Event Data 2 values as described in Table 59. |
| 16 | Event Data 3 | Bitmap of the CPU that causes the system CATERR.<br>[0]: CPU1<br>[1]: CPU2<br>**Note**: If more than one bit is set, the BMC cannot determine the source of the CATERR. |

**Table 59. Internal Error Sensor – Event Data 2 Values – Next Steps**

| ED2 | Description | Next Steps |
|---|---|---|
| 00h | Unknown | 1. Cross test the processors.<br>2. Replace the processors depending on the results of the test. |
| 01h | CATERR | This error is typically caused by other platform components.<br>1. Check for other errors near the time of the CATERR event.<br>2. Verify all peripherals are plugged in and operating correctly, particularly hard drives, optical drives, and I/O.<br>3. Update system firmware and drivers. |
| 2h | CPU Core Error | 1. Cross test the processors.<br>2. Replace the processors depending on the results of the test. |
| 3h | MSID Mismatch | Verify the processor is supported by the baseboard. Check the board *Technical Product Specification (TPS).* |

## 6.3 CPU Missing Sensor

The CPU Missing sensor is a discrete sensor that only reports if the processor is not installed. The most common instance of this event is due to a processor populated in an incorrect socket.

**Table 60. CPU Missing Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| 11 | Sensor Type | 07h = Processor |
| 12 | Sensor Number | 82h |
| 13 | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 03h (Digital Discrete) |
| 14 | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset = 1h (State Asserted) |
| 15 | Event Data 2 | Not used |
| 16 | Event Data 3 | Not used |

### 6.3.1 CPU Missing Sensor – Next Steps

The following steps are used to correct the fault condition and clear the sensor state:

1. AC power down the server.
2. Install the missing processor into the correct slot.
3. AC power on the server.

## 6.4 Intel® UPI Interconnect Sensors

The Intel UPI Link Width Reduced sensor is used by the BIOS POST to report when the link width has been reduced. Therefore, the Generator ID is 01h.

The Intel UPI Error sensors are reported by the BIOS SMI Handler to the BMC, so the Generator ID is 33h.

### 6.4.1 Intel® UPI Link Width Reduced Sensor

BIOS POST has reduced the Intel® UPI Link Width because of an error condition seen during initialization.

**Table 61. Intel® UPI Link Width Reduced Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| 8, 9 | Generator ID | 0001h = BIOS POST |
| 11 | Sensor Type | 13h = Critical Interrupt |
| 12 | Sensor Number | 09h |
| 13 | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 77h (OEM Discrete) |
| 14 | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset<br>1h = Reduced to ½ width<br>2h = Reduced to ¼ width |
| 15 | Event Data 2 | 0–3 = CPU0-3 |
| 16 | Event Data 3 | Not used |

#### 6.4.1.1 Intel® UPI Link Width Reduced Sensor – Next Steps

If the error continues:

1. Check if the processor is installed correctly.
2. Inspect the socket for bent pins.
3. Cross test the processor. If the issue remains with the processor socket, replace the main board; otherwise replace the processor.

### 6.4.2 Intel® UPI Correctable Error Sensor

The system detected an error and corrected it. This is an informational event.

**Table 62. Intel UPI® Correctable Error Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 0033h = BIOS SMI Handler |
| **11** | Sensor Type | 13h = Critical Interrupt |
| **12** | Sensor Number | 06h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 72h (OEM Discrete) |
| **14** | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset = Reserved |
| **15** | Event Data 2 | 0–3 = CPU1-4 |
| **16** | Event Data 3 | Not used |

#### 6.4.2.1 Intel® UPI Correctable Error Sensor – Next Steps

This is an Informational event only. Correctable errors are acceptable and normal at a low rate of occurrence. If the error continues:

1. Check if the processor is installed correctly.
2. Inspect the socket for bent pins.
3. Cross test the processor. If the issue remains with the processor socket, replace the main board, otherwise the processor.

## 6.5 Processor ERR2 Timeout Sensor

The BMC supports an ERR2 Timeout Sensor that asserts if any CPU's ERR2 signal has been asserted for longer than a fixed time period (> 90 seconds). ERR[2] is a processor signal that indicates when the IIO (Integrated IO module in the processor) has a fatal error which could not be communicated to the core to trigger SMI. ERR[2] events are fatal error conditions, where the BIOS and OS will attempt to gracefully handle error, but may not always do so reliably. A continuously asserted ERR2 signal is an indication that the BIOS cannot service the condition that caused the error. This is usually because that condition prevents the BIOS from running.

When an ERR2 timeout occurs, the BMC asserts/deasserts the ERR2 Timeout Sensor, and logs a SEL event for that sensor. The default behavior for BMC core firmware is to initiate a system reset upon detection of an ERR2 timeout. The BIOS setup utility provides an option to disable or enable system reset by the BMC on detection of this condition.

**Table 63. Processor ERR2 Timeout Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | `07h` = Processor |
| **12** | Sensor Number | `7Ch` = Processor ERR2 Timeout |
| **13** | Event Direction and Event Type | [7] Event direction<br>`0b` = Assertion Event<br>`1b` = Deassertion Event<br>[6:0] Event Type = `03h` ("digital" discrete) |
| **14** | Event Data 1 | [7:6] – `10b` = OEM code in Event Data 2<br>[5:4] – `00b` = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset = `1h` (State Asserted) |
| **15** | Event Data 2 | [7:4] – Not used<br>[3:0] – Bitmap of the CPU that causes the ERR2 timeout.<br>[0]: CPU1<br>[1]: CPU2 |
| **16** | Event Data 3 | Not used |

### 6.5.1 Processor ERR2 Timeout – Next Steps

Check the SEL for any other events around the time of the failure.

Take note of all IPMI activity that was occurring around the time of the failure. Capture a System BMC Debug Log as soon as possible after experiencing this failure. This log can be captured from the Integrated BMC Web Console or by using the Syscfg utility (`syscfg/sbmcdl Public filename.zip`). Send the log file to the system manufacturer or Intel representative for failure analysis.

# 7. Memory Subsystem

Intel servers report memory errors, status, and configuration in the SEL.

## 7.1 Memory RAS Configuration Status

A Memory RAS Configuration Status event is logged after an AC power-on occurs, only if any RAS Mode is currently configured, and only if RAS Mode is successfully initiated.

This is to make sure that there is a record in the SEL telling what the RAS Mode was at the time that the system started up. This is only logged after AC power-on, not DC power-on.

The Memory RAS Configuration Status Sensor is also used to log an event during POST whenever there is a RAS configuration error. This is a case where a RAS Mode has been selected but when the system boots, the memory configuration cannot support the RAS Mode. The memory configuration fails, and operates in Independent Channel Mode.

In the SEL record logged, the ED1 Offset value is “RAS Configuration Disabled”, and ED3 contains the RAS Mode that is currently selected but could not be configured. ED2 gives the reason for the RAS configuration failure – at present, only two “RAS Configuration Error Type” values are implemented:

0 = None – This is used for an AC power-on log record when the RAS configuration is successfully configured.

3 = Invalid DIMM Configuration for RAS Mode – The installed DIMM configuration cannot support the currently selected RAS Mode. This may be due to DIMMs that have failed or been disabled, so when this reason has been logged, the user should check the preceding SEL events to see whether there are DIMM error events.

**Table 64. Memory RAS Configuration Status Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 0001h = BIOS POST |
| **11** | Sensor Type | 0Ch = Memory |
| **12** | Sensor Number | 02h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 09h (digital Discrete) |
| **14** | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset as described in Table 65 |
| **15** | Event Data 2 | RAS Configuration Error Type<br>[7:4] = Reserved<br>[3:0] = Configuration Error<br>0 = None<br>3 = Invalid DIMM Configuration for RAS Mode<br>All other values are reserved. |
| **16** | Event Data 3 | RAS Mode Configured<br>[7:4] = Reserved<br>[3:0] = RAS Mode<br>0h = None (Independent Channel Mode)<br>1h = Mirroring Mode<br>4h = Rank Sparing Mode |

**Table 65. Memory RAS Configuration Status Sensor – Event Trigger Offset – Next Steps**

| Event Trigger Offset | | Description | Next Steps |
|---|---|---|---|
| 01h | RAS configuration enabled. | User enabled mirrored channel mode in setup. | Informational event only. |
| 00h | RAS configuration disabled. | Mirrored channel mode is disabled (either in setup or due to unavailability of memory at post, in which case post error 8500 is also logged). | • If this event is accompanied by a post error 8500, there was a problem applying the mirroring configuration to the memory. Check for other errors related to the memory and troubleshoot accordingly.<br>• If there is no post error, mirror mode was simply disabled in BIOS setup and this should be considered informational only. |

## 7.2 Memory RAS Mode Select

Memory RAS Mode Select events are logged to record changes in RAS Mode.

When a RAS Mode selection is made that changes the RAS Mode (including selecting a RAS Mode from or to Independent Channel Mode), that change is logged to SEL in a Memory RAS Mode Select event message, which records the previous RAS Mode (from) and the newly selected RAS Mode (to). The event also includes an Offset value in ED1, which indicates whether the mode change left the system with a RAS Mode active (Enabled), or not (Disabled – Independent Channel Mode selected). This sensor provides the Spare Channel mode RAS Configuration status. Memory RAS Mode Select is an informational event.

**Table 66. Memory RAS Mode Select Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 0001h = BIOS POST |
| **11** | Sensor Type | 0Ch = Memory |
| **12** | Sensor Number | 12h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 09h (digital Discrete) |
| **14** | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset<br>0h = RAS Configuration Disabled<br>1h = RAS Configuration Enabled |
| **15** | Event Data 2 | Prior RAS Mode<br>[7:4] = Reserved<br>[3:0] = RAS Mode<br>0h = None (Independent Channel Mode)<br>1h = Mirroring Mode<br>4h = Rank Sparing Mode |
| **16** | Event Data 3 | Selected RAS Mode<br>[7:4] = Reserved<br>[3:0] = RAS Mode<br>0h = None (Independent Channel Mode)<br>1h = Mirroring Mode<br>4h = Rank Sparing Mode |

## 7.3 Mirroring Redundancy State

Mirroring Mode protects memory data by full redundancy – keeping complete copies of all data on both channels of a Mirroring Domain (channel pair). If an Uncorrectable Error, which is normally fatal, occurs on one channel of a pair, and the other channel is still intact and operational, then the Uncorrectable Error is "demoted" to a Correctable Error, and the failed channel is disabled. Because the Mirror Domain is no longer redundant, a Mirroring Redundancy State SEL Event is logged.

For different platforms, there are some differences in the event definition. For the following product families, refer to Table 67.

**Table 67. Mirroring Redundancy State Sensor Typical Characteristics For Intel® Server Boards Based On Intel® Server Board Based On The 3rd Gen Intel® Scalable Processor Family**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 0033h = BIOS SMI Handler |
| **11** | Sensor Type | 0Ch = Memory |
| **12** | Sensor Number | 01h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 0Bh (Generic Discrete) |
| **14** | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset<br>0h = Fully Redundant<br>1h = Redundancy Degraded |
| **15** | Event Data 2 | Location<br>[7:4] = DIMM index<br>0–1 = DIMM 1–2, DIMM index per channel<br>[3:0] = Rank index (Physical rank index per DIMM) |
| **16** | Event Data 3 | Location<br>[7:4] = Socket index<br>0–3 = CPU0-3<br>[3:0] = Channel index<br>0–5= Channel A-H, Channel index for socket |

### 7.3.1 Mirroring Redundancy State Sensor – Next Steps

This event is accompanied by memory errors indicating the source of the issue. Troubleshoot accordingly (probably replace affected DIMM).

For boards with DIMM Fault LEDs, the appropriate Fault LED is lit to indicate which DIMM was the source of the error triggering the Mirroring Failover action, that is, the failing DIMM.

## 7.4 Sparing Redundancy State

Rank Sparing Mode is a Memory RAS configuration option that reserves one memory rank per channel as a "spare rank". If any rank on a given channel experiences enough Correctable ECC Errors to cross the Correctable Error Threshold, the data in that rank is copied to the spare rank, and then the spare rank is mapped into the memory array to replace the failing rank.

Rank Sparing Mode protects memory data by reserving a “Spare Rank” on each channel that has memory installed on it. If a Correctable Error Threshold event occurs, the data from the failing rank is copied to the Spare Rank on the same channel, and the failing DIMM is disabled. Because the Sparing Domain is no longer redundant, a Sparing Redundancy State SEL Event is logged.

**Table 68. Sparing Redundancy State Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 0033h = BIOS SMI Handler |
| **11** | Sensor Type | 0Ch = Memory |
| **12** | Sensor Number | 11h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 0Bh (Generic Discrete) |
| **14** | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset<br>0h = Fully Redundant<br>1h = Redundancy Degraded |
| **15** | Event Data 2 | Location<br>[7:4] = DIMM index<br>0–1 = DIMM 1–2, DIMM index per channel<br>[3:0] = Rank index (Physical rank index per DIMM) |
| **16** | Event Data 3 | Location<br>[7:4] = Socket index<br>0–3 = CPU0-3<br>[3:0] = Channel index<br>0–5= Channel A-H, Channel index for socket |

### 7.4.1 Sparing Redundancy State Sensor – Next Steps

This event is accompanied by memory errors indicating the source of the issue. Troubleshoot accordingly (probably replace affected DIMM).

For boards with DIMM Fault LEDs, the appropriate Fault LED is lit to indicate which DIMM was the source of the error triggering the Mirroring Failover action, that is, the failing DIMM.

## 7.5 Memory RAS ADC/ADDDC Mode

ADDDC memory RAS supports x4 DIMM and is enabled by default when x4 DIMM plugged in. ADDDC can do bank level VLS (Virtual Lockstep), and rank VLS. ADDDC works on channel level granularity. On single rank, it has a maximum tolerance of three hard errors. There are several factors that impact VLS create. Rank number on channel, ADDDC region number, and failed error location are the three most likely factors. ICX standard RAS SKU has one ADDDDC region per channel and can only support bank level VLS. Advance RAS SKU has two ADDDC regions per channel and can support bank, rank level VLS.

**Table 69. Memory ADDDC Mode Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 0033h = BIOS SMI Handler |
| **11** | Sensor Type | 0Ch = Memory |
| **12** | Sensor Number | 20h |

| Byte | Field | Description |
|---|---|---|
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Generic Discrete) |
| **14** | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset<br>0h = Fully Redundant<br>1h = Redundancy Degraded |
| **15** | Event Data 2 | Location<br>[7:4] = DIMM index<br>0–1 = DIMM 1–2, DIMM index per channel<br>[3:0] = Rank index (Physical rank index per DIMM) |
| **16** | Event Data 3 | Location<br>[7:4] = Socket index<br>0–3 = CPU0-3<br>[3:0] = Channel index<br>0–5= Channel A-H, Channel index for socket |

## 7.6 ECC and Address Parity

1. Memory data errors are logged as correctable or uncorrectable.
2. Uncorrectable errors are fatal.
3. Memory addresses are protected with parity bits and a parity error is logged. This is a fatal error.

### 7.6.1 Memory Correctable and Uncorrectable ECC Error

ECC errors are divided into Uncorrectable ECC Errors and Correctable ECC Errors. A “Correctable ECC Error” actually represents a threshold overflow. More Correctable Errors are detected at the memory controller level for a given DIMM within a given timeframe. In both cases, the error can be narrowed down to particular DIMM(s). The BIOS SMI error handler uses this information to log the data to the BMC SEL and identify the failing DIMM module.

**Table 70. Correctable And Uncorrectable ECC Error Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 0033h = BIOS SMI Handler |
| **11** | Sensor Type | 0Ch = Memory |
| **12** | Sensor Number | 02h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| **14** | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset as described in table 81 |
| **15** | Event Data 2 | [7:4] = DIMM index<br>0–1 = DIMM1-2, DIMM index per Channel<br>[3:0] = Physical rank index per DIMM. |
| **16** | Event Data 3 | [7:4] = Socket index<br>0–3= CPU0-3<br>[3:0] = Channel index<br>0–5 = Channel A-H Channel index for Socket |

**Table 71. Correctable And Uncorrectable ECC Error Sensor Event Trigger Offset – Next Steps**

| Event Trigger Offset | | Description | Next Steps |
|---|---|---|---|
| 01h | Uncorrectable ECC Error | An uncorrectable (multi-bit) ECC error has occurred. This is a fatal issue that will typically lead to an OS crash (unless memory has been configured in a RAS mode). The system will generate a CATERR# (catastrophic error) and an MCE (Machine Check Exception Error).<br>While the error may be due to a failing DRAM chip on the DIMM, it can also be caused by incorrect seating or improper contact between socket and DIMM, or by bent pins in the processor socket. | 1. If needed, decode DIMM location from hex version of SEL.<br>2. Verify the DIMM is seated properly.<br>3. Examine gold fingers on edge of the DIMM to verify contacts are clean.<br>4. Inspect the processor socket this DIMM is connected to for bent pins, and if found, replace the board.<br>5. Consider replacing the DIMM as a preventative measure. For multiple occurrences, replace the DIMM. |
| 00h | Correctable ECC Error threshold reached | There have been too many (10 or more) correctable ECC errors for this particular DIMM since last boot. This event in itself does not pose any direct problems because the ECC errors are still being corrected. Depending on the RAS configuration of the memory, the IMC may take the affected DIMM offline. | Even though this event doesn't immediately lead to problems, it can indicate one of the DIMM modules is slowly failing. If this error occurs more than once:<br>1. If needed, decode DIMM location from hex version of SEL.<br>2. Verify the DIMM is seated properly.<br>3. Examine gold fingers on edge of the DIMM to verify contacts are clean.<br>4. Inspect the processor socket this DIMM is connected to for bent pins, and if found, replace the board.<br>5. Consider replacing the DIMM as a preventative measure. For multiple occurrences, replace the DIMM. |

### 7.6.2 Memory Address Parity Error

Address Parity errors are errors detected in the memory addressing hardware. Because these affect the addressing of memory contents, they can potentially lead to the same sort of failures as ECC errors. They are logged as a distinct type of error because they affect memory addressing rather than memory contents, but otherwise they are treated exactly the same as Uncorrectable ECC Errors. Address Parity errors are logged to the BMC SEL, with Event Data to identify the failing address by channel and DIMM to the extent that it is possible to do so.

For different platforms, there are some differences in the event definition. For the following product families, refer to Table 72.

**Table 72. Address Parity Error Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| 8, 9 | Generator ID | 0033h = BIOS SMI Handler |
| 11 | Sensor Type | 0Ch = Memory |
| 12 | Sensor Number | 13h |
| 13 | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| 14 | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset<br>3h = Command and Address Parity Error |

| Byte | Field | Description |
|---|---|---|
| **15** | Event Data 2 | [7:4] = Dimm index<br>0–1 = DIMM1-2, DIMM index per Channel<br>[3:0] = Physical rank index per DIMM |
| **16** | Event Data 3 | [7:4] = Socket index<br>0–3= CPU0-3<br>[3:0] = Channel index<br>0–5 = Channel A-H, Channel index for Socket |

#### 7.6.2.1 Memory Address Parity Error Sensor – Next Steps

These are bit errors that are detected in the memory addressing hardware. An Address Parity Error implies that the memory address transmitted to the DIMM addressing circuitry has been compromised, and data read or written is compromised in turn. An Address Parity Error is logged as such in SEL but in all other ways, is treated the same as an Uncorrectable ECC Error.

While the error may be due to a failing DRAM chip on the DIMM, it can also be caused by incorrect seating or improper contact between the socket and DIMM, or by the bent pins in the processor socket.

1. If needed, decode DIMM location from hex version of SEL.
2. Verify the DIMM is seated properly.
3. Examine gold fingers on edge of the DIMM to verify contacts are clean.
4. Inspect the processor socket this DIMM is connected to for bent pins, and if found, replace the board.
5. Consider replacing the DIMM as a preventative measure. For multiple occurrences, replace the DIMM.

# 8. PCI Express* and Legacy PCI Subsystem

The *PCI Express* (PCIe*) Specification* defines standard error types under the Advanced Error Reporting (AER) capabilities. The BIOS logs AER events into the SEL.

The *Legacy PCI Specification* error types are parity error (PERR) and system error (SERR). These errors are supported and logged into the SEL.

## 8.1 Legacy PCI Errors

Legacy PCI errors include PERR and SERR; both are fatal errors.

**Table 73. Legacy PCI Error Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 0033h = BIOS SMI Handler |
| **11** | Sensor Type | 13h = Critical Interrupt |
| **12** | Sensor Number | 03h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| **14** | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset<br>4h = PCI PERR<br>5h = PCI SERR |
| **15** | Event Data 2 | PCI Bus number |
| **16** | Event Data 3 | [7:3] – PCI Device number<br>[2:0] – PCI Function number |

### 8.1.1 Legacy PCI Error Sensor – Next Steps

1. Decode the bus, device, and function to identify the card.
2. If this is an add-in card:
   a. Verify the card is inserted properly.
   b. Install the card in another slot and check whether the error follows the card or stays with the slot.
   c. Update all firmware and drivers, including non-Intel components.
3. If this is an on-board device:
   a. Update all BIOS, firmware, and drivers.
   b. Replace the board.

## 8.2 PCIe* Errors

PCIe* error events are either correctable (informational event) or fatal. In both cases information is logged to help identify the source of the PCIe* error and the bus, device, and function is included in the extended data fields. The PCIe* devices are mapped in the operating system by bus, device, and function. Each device is uniquely identified by the bus, device, and function. PCIe* device information can be found in the operating system.

### 8.2.1 PCIe* Fatal Errors and Fatal Error #2

When a PCIe* fatal error is reported to the BIOS SMI handler, it records the error as shown in Table 74.

**Table 74. PCIe* Fatal Error Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 0033h = BIOS SMI Handler |
| **11** | Sensor Type | 13h = Critical Interrupt |
| **12** | Sensor Number | 04h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 70h (OEM Specific) |
| **14** | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger<br>0h = Data Link Layer Protocol Error<br>1h = Surprise Link Down Error<br>2h = Completer Abort<br>3h = Unsupported Request<br>4h = Poisoned TLP<br>5h = Flow Control Protocol<br>6h = Completion Timeout<br>7h = Receiver Buffer Overflow<br>8h = ACS Violation<br>9h = Malformed TLP<br>Ah = ECRC Error<br>Bh = Received Fatal Message From Downstream<br>Ch = Unexpected Completion<br>Dh = Received ERR_NONFATAL Message<br>Eh = Uncorrectable Internal<br>Fh = MC Blocked TLP |
| **15** | Event Data 2 | PCI Bus number |
| **16** | Event Data 3 | [7:3] – PCI Device number<br>[2:0] – PCI Function number |

The PCIe* Fatal Error #2 is a continuation of the PCIe* Fatal Error.

**Table 75. Pcie* Fatal Error #2 Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 0033h = BIOS SMI Handler |
| **11** | Sensor Type | 13h = Critical Interrupt |
| **12** | Sensor Number | 14h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 76h (OEM Specific) |
| **14** | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset<br>0h = Atomic Egress Blocked<br>1h = TLP Prefix Blocked<br>Fh = Unspecified Non-AER Fatal Error |
| **15** | Event Data 2 | PCI Bus number |
| **16** | Event Data 3 | [7:3] – PCI Device number<br>[2:0] – PCI Function number |

#### 8.2.1.1 PCIe* Fatal Error and Fatal Error #2 Sensor – Next Steps

1. Decode the bus, device, and function to identify the card.
2. If this is an add-in card:
   a. Verify the card is inserted properly.
   b. Install the card in another slot and check if the error follows the card or stays with the slot.
   c. Update all firmware and drivers, including non-Intel components.
3. If this is an on-board device:
   a. Update all BIOS, firmware, and drivers.
   b. Replace the board.

### 8.2.2 PCIe* Correctable Errors

When a PCIe* correctable error is reported to the BIOS SMI handler, it records the error as shown in Table 76.

**Table 76. Pcie* Correctable Error Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 0033h = BIOS SMI Handler |
| **11** | Sensor Type | 13h = Critical Interrupt |
| **12** | Sensor Number | 05h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 71h (OEM Specific) |
| **14** | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset<br>0h = Receiver Error<br>1h = Bad DLLP<br>2h = Bad TLP<br>3h = Replay Num Rollover<br>4h = Replay Timer timeout<br>5h = Advisory Non-fatal<br>6h = Link BW Changed<br>7h = Correctable Internal<br>8h = Header Log Overflow |
| **15** | Event Data 2 | PCI Bus number |
| **16** | Event Data 3 | [7:3] – PCI Device number<br>[2:0] – PCI Function number |

#### 8.2.2.1 PCIe* Correctable Error Sensor – Next Steps

This is an informational event only. Correctable errors are acceptable and normal at a low rate of occurrence. If the error continues:

1. Decode the bus, device, and function to identify the card.
2. If this is an add-in card:
   a. Verify the card is inserted properly.
   b. Install the card in another slot and check if the error follows the card or stays with the slot.
   c. Update all firmware and drivers, including non-Intel components.
3. If this is an on-board device:
   a. Update all BIOS, firmware, and drivers.
   b. Replace the board.

### 8.2.3 IIO Internal Error

In additional to PCIe* Root Port, there are several components inside IIO Subsystem, such as VTD, CBDMA, M2PCIE, IRP, RING, ITC, OTC, etc. This error records the error as shown in Table 85:

**Table 77. IIO Internal Error Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 0033h = BIOS SMI Handler |
| **11** | Sensor Type | 19h = IIO Internal Error |
| **12** | Sensor Number | 08h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 75h (OEM Discrete) |
| **14** | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset<br>0h = VTD<br>1h = CBDMA<br>2h = M2PCIE<br>3h = IRP<br>4h = RING<br>5h = ITC<br>6h = OTC |
| **15** | Event Data 2 | PCI Bus number |
| **16** | Event Data 3 | [7:3] – PCI Device number<br>[2:0] – PCI Function number |

# 9. System BIOS Events

There are a number of events that are owned by the system BIOS. These events can occur during Power On Self Test (POST) or when coming out of a sleep state. Not all of these events signify errors. Some events are described in other chapters in this document (for example, memory events).

## 9.1 System Boot

At the end of POST, just before the actual OS boot occurs, a System Boot Event is logged. This basically serves to mark the transition of control from completed POST to OS Loader. It is an information only event.

**Table 78. System Event Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 0001h = BIOS POST |
| **11** | Sensor Type | 12h = System Event |
| **12** | Sensor Number | 83h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset<br>01h = System Boot |
| **15** | Event Data 2 | Not used |
| **16** | Event Data 3 | Not used |

## 9.2 System Firmware Progress (Formerly POST Error)

The BIOS logs any POST errors to the SEL. The two-byte POST code gets logged in the ED2 and ED3 bytes in the SEL entry. This event is not logged every time a POST error is displayed. Even though this event indicates an error, it may not be a fatal error. If this is a serious error, there is typically also a corresponding SEL entry logged for whatever was the cause of the error; this event may contain more information about what happened than the POST error event.

**Table 79. POST Error Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| 8, 9 | Generator ID | 0001h = BIOS POST |
| 11 | Sensor Type | 0Fh = System Firmware Progress (formerly POST Error) |
| 12 | Sensor Number | 06h |
| 13 | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| 14 | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset = 0h |
| 15 | Event Data 2 | Low Byte of POST Error Code |
| 16 | Event Data 3 | High Byte of POST Error Code |

### 9.2.1 System Firmware Progress (Formerly POST Error) – Next Steps

See the following table for POST Error Codes.

**Table 80. POST Error Codes**

| Error Code | Error Message | Action message | Response |
|---|---|---|---|
| **0012** | System RTC date/time not set | | Major |
| **0048** | Password check failed | Put right password. | Major |
| **0140** | PCI component encountered a PERR error | | Major |
| **0141** | PCI resource conflict | | Major |
| **0146** | PCI out of resources error | Enable Memory Mapped I/O above 4 GB item at SETUP to use 64-bit MMIO. | Major |
| **0191** | Processor core/thread count mismatch detected | Use identical CPU type. | Fatal |
| **0192** | Processor cache size mismatch detected | Use identical CPU type. | Fatal |
| **0194** | Processor family mismatch detected | Use identical CPU type. | Fatal |
| **0195** | Processor Intel(R) UPI link frequencies unable to synchronize | | Fatal |
| **0196** | Processor model mismatch detected | Use identical CPU type. | Fatal |
| **0197** | Processor frequencies unable to synchronize | Use identical CPU type. | Fatal |
| **5220** | BIOS Settings reset to default settings | | Major |
| **5221** | Passwords cleared by jumper | | Major |
| **5224** | Password clear jumper is Set | Recommend reminding user to install BIOS password as BIOS admin password is the master keys for several BIOS security features. | Major |
| **8130** | CPU0 disabled | | Major |
| **8131** | CPU1 disabled | | Major |
| **8160** | CPU0 unable to apply microcode update | | Major |
| **8161** | CPU1 unable to apply microcode update | | Major |
| **8170** | CPU0 failed Self-Test (BIST) | | Major |
| **8171** | CPU1 failed Self-Test (BIST) | | Major |
| **8180** | CPU0 microcode update not found | | Minor |
| **8181** | CPU1 microcode update not found | | Minor |
| **8190** | Watchdog timer failed on last boot. | | Major |
| **8198** | OS boot watchdog timer failure. | | Major |
| **8300** | Baseboard Management Controller failed self-test. | | Major |
| **8305** | Hot Swap Controller failure | | Major |
| **83A0** | Management Engine (ME) failed self-test. | | Major |
| **83A1** | Management Engine (ME) Failed to respond. | | Major |
| **84F2** | Baseboard management controller failed to respond | | Major |
| **84F3** | Baseboard Management Controller in Update Mode. | | Major |
| **84F4** | Baseboard Management Controller Sensor Data Record empty. | Update right SDR. | Major |
| **84FF** | System Event Log full | Clear SEL through EWS or SELVIEW utility. | Minor |
| **85FC** | Memory component could not be configured in the selected RAS mode | | Major |
| **8501** | Memory Population Error | Plug DIMM at right population. | Major |

| Error Code | Error Message | Action message | Response |
|---|---|---|---|
| **8502** | PMem invalid DIMM population found on the system. | Populate valid POR pmem DIMM population. | Major |
| **8520** | Memory failed test/initialization CPU0_DIMM_A1 | Remove the disabled DIMM. | Major |
| **8521** | Memory failed test/initialization CPU0_DIMM_A2 | Remove the disabled DIMM. | Major |
| **8522** | Memory failed test/initialization CPU0_DIMM_A3 | Remove the disabled DIMM. | Major |
| **8523** | Memory failed test/initialization CPU0_DIMM_B1 | Remove the disabled DIMM. | Major |
| **8524** | Memory failed test/initialization CPU0_DIMM_B2 | Remove the disabled DIMM. | Major |
| **8525** | Memory failed test/initialization CPU0_DIMM_B3 | Remove the disabled DIMM. | Major |
| **8526** | Memory failed test/initialization CPU0_DIMM_C1 | Remove the disabled DIMM. | Major |
| **8527** | Memory failed test/initialization CPU0_DIMM_C2 | Remove the disabled DIMM. | Major |
| **8528** | Memory failed test/initialization CPU0_DIMM_C3 | Remove the disabled DIMM. | Major |
| **8529** | Memory failed test/initialization CPU0_DIMM_D1 | Remove the disabled DIMM. | Major |
| **852A** | Memory failed test/initialization CPU0_DIMM_D2 | Remove the disabled DIMM. | Major |
| **852B** | Memory failed test/initialization CPU0_DIMM_D3 | Remove the disabled DIMM. | Major |
| **852C** | Memory failed test/initialization CPU0_DIMM_E1 | Remove the disabled DIMM. | Major |
| **852D** | Memory failed test/initialization CPU0_DIMM_E2 | Remove the disabled DIMM. | Major |
| **852E** | Memory failed test/initialization CPU0_DIMM_E3 | Remove the disabled DIMM. | Major |
| **852F** | Memory failed test/initialization CPU0_DIMM_F1 | Remove the disabled DIMM. | Major |
| **8530** | Memory failed test/initialization CPU0_DIMM_F2 | Remove the disabled DIMM. | Major |
| **8531** | Memory failed test/initialization CPU0_DIMM_F3 | Remove the disabled DIMM. | Major |
| **8532** | Memory failed test/initialization CPU0_DIMM_G1 | Remove the disabled DIMM. | Major |
| **8533** | Memory failed test/initialization CPU0_DIMM_G2 | Remove the disabled DIMM. | Major |
| **8534** | Memory failed test/initialization CPU0_DIMM_G3 | Remove the disabled DIMM. | Major |
| **8535** | Memory failed test/initialization CPU0_DIMM_H1 | Remove the disabled DIMM. | Major |
| **8536** | Memory failed test/initialization CPU0_DIMM_H2 | Remove the disabled DIMM. | Major |
| **8537** | Memory failed test/initialization CPU0_DIMM_H3 | Remove the disabled DIMM. | Major |
| **8538** | Memory failed test/initialization CPU1_DIMM_A1 | Remove the disabled DIMM. | Major |
| **8539** | Memory failed test/initialization CPU1_DIMM_A2 | Remove the disabled DIMM. | Major |
| **853A** | Memory failed test/initialization CPU1_DIMM_A3 | Remove the disabled DIMM. | Major |
| **853B** | Memory failed test/initialization CPU1_DIMM_B1 | Remove the disabled DIMM. | Major |
| **853C** | Memory failed test/initialization CPU1_DIMM_B2 | Remove the disabled DIMM. | Major |
| **853D** | Memory failed test/initialization CPU1_DIMM_B3 | Remove the disabled DIMM. | Major |
| **853E** | Memory failed test/initialization CPU1_DIMM_C1 | Remove the disabled DIMM. | Major |
| **853F (Go to 85C0)** | Memory failed test/initialization CPU1_DIMM_C2 | Remove the disabled DIMM. | Major |
| **8540** | Memory disabled.CPU0_DIMM_A1 | Remove the disabled DIMM. | Major |
| **8541** | Memory disabled.CPU0_DIMM_A2 | Remove the disabled DIMM. | Major |
| **8542** | Memory disabled.CPU0_DIMM_A3 | Remove the disabled DIMM. | Major |
| **8543** | Memory disabled.CPU0_DIMM_B1 | Remove the disabled DIMM. | Major |
| **8544** | Memory disabled.CPU0_DIMM_B2 | Remove the disabled DIMM. | Major |
| **8545** | Memory disabled.CPU0_DIMM_B3 | Remove the disabled DIMM. | Major |
| **8546** | Memory disabled.CPU0_DIMM_C1 | Remove the disabled DIMM. | Major |
| **8547** | Memory disabled.CPU0_DIMM_C2 | Remove the disabled DIMM. | Major |
| **8548** | Memory disabled.CPU0_DIMM_C3 | Remove the disabled DIMM. | Major |
| **8549** | Memory disabled.CPU0_DIMM_D1 | Remove the disabled DIMM. | Major |
| **854A** | Memory disabled.CPU0_DIMM_D2 | Remove the disabled DIMM. | Major |

| Error Code | Error Message | Action message | Response |
|---|---|---|---|
| **854B** | Memory disabled.CPU0_DIMM_D3 | Remove the disabled DIMM. | Major |
| **854C** | Memory disabled.CPU0_DIMM_E1 | Remove the disabled DIMM. | Major |
| **854D** | Memory disabled.CPU0_DIMM_E2 | Remove the disabled DIMM. | Major |
| **854E** | Memory disabled.CPU0_DIMM_E3 | Remove the disabled DIMM. | Major |
| **854F** | Memory disabled.CPU0_DIMM_F1 | Remove the disabled DIMM. | Major |
| **8550** | Memory disabled.CPU0_DIMM_F2 | Remove the disabled DIMM. | Major |
| **8551** | Memory disabled.CPU0_DIMM_F3 | Remove the disabled DIMM. | Major |
| **8552** | Memory disabled.CPU0_DIMM_G1 | Remove the disabled DIMM. | Major |
| **8553** | Memory disabled.CPU0_DIMM_G2 | Remove the disabled DIMM. | Major |
| **8554** | Memory disabled.CPU0_DIMM_G3 | Remove the disabled DIMM. | Major |
| **8555** | Memory disabled.CPU0_DIMM_H1 | Remove the disabled DIMM. | Major |
| **8556** | Memory disabled.CPU0_DIMM_H2 | Remove the disabled DIMM. | Major |
| **8557** | Memory disabled.CPU0_DIMM_H3 | Remove the disabled DIMM. | Major |
| **8558** | Memory disabled.CPU1_DIMM_A1 | Remove the disabled DIMM. | Major |
| **8559** | Memory disabled.CPU1_DIMM_A2 | Remove the disabled DIMM. | Major |
| **855A** | Memory disabled.CPU1_DIMM_A3 | Remove the disabled DIMM. | Major |
| **855B** | Memory disabled.CPU1_DIMM_B1 | Remove the disabled DIMM. | Major |
| **855C** | Memory disabled.CPU1_DIMM_B2 | Remove the disabled DIMM. | Major |
| **855D** | Memory disabled.CPU1_DIMM_B3 | Remove the disabled DIMM. | Major |
| **855E** | Memory disabled.CPU1_DIMM_C1 | Remove the disabled DIMM. | Major |
| **855F (Go to 85D0)** | Memory disabled.CPU1_DIMM_C2 | Remove the disabled DIMM. | Major |
| **8560** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_A1 | | Major |
| **8561** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_A2 | | Major |
| **8562** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_A3 | | Major |
| **8563** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_B1 | | Major |
| **8564** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_B2 | | Major |
| **8565** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_B3 | | Major |
| **8566** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_C1 | | Major |
| **8567** | Memory encountered a Serial Presence Detection (SPD) failure.CPU0_DIMM_C2 | | Major |
| **8568** | Memory encountered a Serial Presence Detection (SPD) failure.CPU0_DIMM_C3 | | Major |
| **8569** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_D1 | | Major |
| **856A** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_D2 | | Major |
| **856B** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_D3 | | Major |
| **856C** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_E1 | | Major |

| Error Code | Error Message | Action message | Response |
|---|---|---|---|
| **856D** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_E2 | | Major |
| **856E** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_E3 | | Major |
| **856F** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_F1 | | Major |
| **8570** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_F2 | | Major |
| **8571** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_F3 | | Major |
| **8572** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_G1 | | Major |
| **8573** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_G2 | | Major |
| **8574** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_G3 | | Major |
| **8575** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_H1 | | Major |
| **8576** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_H2 | | Major |
| **8577** | Memory encountered a Serial Presence Detection(SPD) failure.CPU0_DIMM_H3 | | Major |
| **8578** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_A1 | | Major |
| **8579** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_A2 | | Major |
| **857A** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_A3 | | Major |
| **857B** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_B1 | | Major |
| **857C** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_B2 | | Major |
| **857D** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_B3 | | Major |
| **857E** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_C1 | | Major |
| **857F (Go to 85E0)** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_C2 | | Major |
| **85C0** | Memory failed test/initialization CPU1_DIMM_C3 | Remove the disabled DIMM. | Major |
| **85C1** | Memory failed test/initialization CPU1_DIMM_D1 | Remove the disabled DIMM. | Major |
| **85C2** | Memory failed test/initialization CPU1_DIMM_D2 | Remove the disabled DIMM. | Major |
| **85C3** | Memory failed test/initialization CPU1_DIMM_D3 | Remove the disabled DIMM. | Major |
| **85C4** | Memory failed test/initialization CPU1_DIMM_E1 | Remove the disabled DIMM. | Major |
| **85C5** | Memory failed test/initialization CPU1_DIMM_E2 | Remove the disabled DIMM. | Major |
| **85C6** | Memory failed test/initialization CPU1_DIMM_E3 | Remove the disabled DIMM. | Major |
| **85C7** | Memory failed test/initialization CPU1_DIMM_F1 | Remove the disabled DIMM. | Major |
| **85C8** | Memory failed test/initialization CPU1_DIMM_F2 | Remove the disabled DIMM. | Major |
| **85C9** | Memory failed test/initialization CPU1_DIMM_F3 | Remove the disabled DIMM. | Major |
| **85CA** | Memory failed test/initialization CPU1_DIMM_G1 | Remove the disabled DIMM. | Major |
| **85CB** | Memory failed test/initialization CPU1_DIMM_G2 | Remove the disabled DIMM. | Major |

| Error Code | Error Message | Action message | Response |
|---|---|---|---|
| **85CC** | Memory failed test/initialization CPU1_DIMM_G3 | Remove the disabled DIMM. | Major |
| **85CD** | Memory failed test/initialization CPU1_DIMM_H1 | Remove the disabled DIMM. | Major |
| **85CE** | Memory failed test/initialization CPU1_DIMM_H2 | Remove the disabled DIMM. | Major |
| **85CF** | Memory failed test/initialization CPU1_DIMM_H3 | Remove the disabled DIMM. | Major |
| **85D0** | Memory disabled.CPU1_DIMM_C3 | Remove the disabled DIMM. | Major |
| **85D1** | Memory disabled.CPU1_DIMM_D1 | Remove the disabled DIMM. | Major |
| **85D2** | Memory disabled.CPU1_DIMM_D2 | Remove the disabled DIMM. | Major |
| **85D3** | Memory disabled.CPU1_DIMM_D3 | Remove the disabled DIMM. | Major |
| **85D4** | Memory disabled.CPU1_DIMM_E1 | Remove the disabled DIMM. | Major |
| **85D5** | Memory disabled.CPU1_DIMM_E2 | Remove the disabled DIMM. | Major |
| **85D6** | Memory disabled.CPU1_DIMM_E3 | Remove the disabled DIMM. | Major |
| **85D7** | Memory disabled.CPU1_DIMM_F1 | Remove the disabled DIMM. | Major |
| **85D8** | Memory disabled.CPU1_DIMM_F2 | Remove the disabled DIMM. | Major |
| **85D9** | Memory disabled.CPU1_DIMM_F3 | Remove the disabled DIMM. | Major |
| **85DA** | Memory disabled.CPU1_DIMM_G1 | Remove the disabled DIMM. | Major |
| **85DB** | Memory disabled.CPU1_DIMM_G2 | Remove the disabled DIMM. | Major |
| **85DC** | Memory disabled.CPU1_DIMM_G3 | Remove the disabled DIMM. | Major |
| **85DD** | Memory disabled.CPU1_DIMM_H1 | Remove the disabled DIMM. | Major |
| **85DE** | Memory disabled.CPU1_DIMM_H2 | Remove the disabled DIMM. | Major |
| **85DF** | Memory disabled.CPU1_DIMM_H3 | Remove the disabled DIMM. | Major |
| **85E0** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_C3 | | Major |
| **85E1** | Memory encountered a Serial Presence Detection (SPD) failure. CPU1_DIMM_D1 | | Major |
| **85E2** | Memory encountered a Serial Presence Detection (SPD) failure.CPU1_DIMM_D2 | | Major |
| **85E3** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_D3 | | Major |
| **85E4** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_E1 | | Major |
| **85E5** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_E2 | | Major |
| **85E6** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_E3 | | Major |
| **85E7** | Memory encountered a Serial Presence Detection (SPD) failure.CPU1_DIMM_F1 | | Major |
| **85E8** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_F2 | | Major |
| **85E9** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_F3 | | Major |
| **85EA** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_G1 | | Major |
| **85EB** | Memory encountered a Serial Presence Detection (SPD) failure. CPU1_DIMM_G2 | | Major |
| **85EC** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_G3 | | Major |
| **85ED** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_H1 | | Major |

| Error Code | Error Message | Action message | Response |
|---|---|---|---|
| **85EE** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_H2 | | Major |
| **85EF** | Memory encountered a Serial Presence Detection(SPD) failure.CPU1_DIMM_H3 | | Major |
| **8604** | POST Reclaim of non-critical NVRAM variables | | Minor |
| **8605** | BIOS Settings are corrupted | | Major |
| **8606** | NVRAM variable space was corrupted and has been reinitialized | | Major |
| **8607** | Recovery boot has been initiated.<br>Note: The Primary BIOS image may be corrupted or the system may hang during POST. A BIOS update is required. | | Fatal |
| **A100** | BIOS ACM Error | | Major |
| **A421** | PCI component encountered a SERR error | | Fatal |
| **A5A0** | PCI Express component encountered a PERR error | | Minor |
| **A5A1** | PCI Express component encountered an SERR error | | Fatal |
| **A6A0** | DXE Boot Services driver: Not enough memory available to shadow a Legacy Option ROM. | Disable oprom at SETUP to save runtime memory. | Minor |

# 10. Chassis Subsystem

The BMC monitors several aspects of the chassis. Next to logging when the power and reset buttons get pressed, the BMC also monitors chassis intrusion if a chassis intrusion switch is included in the chassis, as well as looking at the network connections, and logging an event whenever the physical network link is lost.

## 10.1 Physical Security

Two sensors are included in the physical security subsystem: chassis intrusion and LAN leash lost.

### 10.1.1 Chassis Intrusion

Chassis Intrusion is monitored on supported chassis, and the BMC logs corresponding events when the chassis lid is opened and closed.

### 10.1.2 LAN Leash Lost

The LAN Leash lost sensor monitors the physical connection on the on-board network ports. If a LAN Leash lost event is logged, this means the network port lost its physical connection.

**Table 81. Physical Security Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 05h = Physical Security |
| **12** | Sensor Number | 04h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset as described in Table 82 |
| **15** | Event Data 2 | Not used |
| **16** | Event Data 3 | Not used |

**Table 82. Physical Security Sensor Event Trigger Offset – Next Steps**

| Event Trigger Offset | | Description | Next Steps |
|---|---|---|---|
| **00h** | Chassis intrusion | The chassis has been opened (or the chassis intrusion sensor is not connected). | 1. Use the Quick Start Guide and the Service Guide to determine whether the chassis intrusion switch is connected properly.<br>2. If this is the case, make sure it makes proper contact when the chassis is closed.<br>3. If this is also the case, someone has opened the chassis. Ensure nobody has access to the system that should not. |
| **04h** | LAN leash lost | A LAN cable that was present when the BMC initialized has been unplugged. This event gets logged when the electrical connection on the NIC connector gets lost. | This is most likely due to unplugging the cable but can also happen if there is an issue with the cable or switch.<br>1. Check the LAN cable and connector for issues.<br>2. Investigate switch logs where possible.<br>3. Ensure nobody has access to the server that should not. |

## 10.2 Front Panel (NMI) Interrupt

The BMC supports a non-maskable interrupt (NMI) sensor for logging an event when a diagnostic interrupt is generated for the following cases:

- The front panel diagnostic interrupt button is pressed.
- The BMC receives an IPMI Chassis Control command that requests this action.

The front panel interrupt button (also referred to as NMI button) is a recessed button on the front panel that allows the user to force a critical interrupt that causes a crash error or kernel panic.

**Table 83. Front Panel (NMI) Interrupt Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 13h = Critical Interrupt |
| **12** | Sensor Number | 05h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset = 0h<br>00 = Front Panel NMI / Diagnostic Interrupt |
| **15** | Event Data 2 | Not used |
| **16** | Event Data 3 | Not used |

### 10.2.1 Front Panel (NMI) Interrupt – Next Steps

The purpose of this button is for diagnosing software issues. When a critical interrupt is generated, the operating system typically saves a memory dump. This allows for exact analysis of what is going on in system memory, which can be useful for software developers, or for troubleshooting the operating system, software, and driver issues.

If this button was not actually pressed, ensure there is no physical fault with the front panel.

This event only gets logged if a user pressed the NMI button or sent an IPMI Chassis Control command requesting this action and, although it causes the operating system to crash, is not an error.

## 10.3 Button Sensor

The BMC logs when the front panel power and reset buttons get pressed. This is purely for informational purposes and these events do not indicate errors.

**Table 84. Button Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 14h = Button/Switch |
| **12** | Sensor Number | 09h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset<br>0h = Power Button<br>2h = Reset Button |
| **15** | Event Data 2 | Not used |
| **16** | Event Data 3 | Not used |

# 11. Miscellaneous Events

The miscellaneous events section addresses sensors not easily grouped with other sensor types.

## 11.1 IPMI Watchdog

Intel® Server Systems support an IPMI watchdog timer that can check to see whether the OS is still responsive. The timer is disabled by default, and has to be enabled manually. It then requires an IPMI-aware utility in the operating system that resets the timer before it expires. If the timer does expire, the BMC can take action if it is configured to do so (reset, power down, power cycle, or generate a critical interrupt).

**Table 85. IPMI Watchdog Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 23h = Watchdog 2 |
| **12** | Sensor Number | 03h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| **14** | Event Data 1 | [7:6] – 11B = Sensor-specific event extension code in Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset as described in Table 86 |
| **15** | Event Data 2 | [7:4] – Interrupt type<br>0h = None<br>1h = SMI<br>2h = NMI<br>3h = Messaging Interrupt<br>Fh = Unspecified<br>All other = Reserved<br>[3:0] – Timer use at expiration<br>0h = Reserved<br>1h = BIOS FRB2<br>2h = BIOS/POST<br>3h = OS Load<br>4h = SMS/OS<br>5h = OEM<br>Fh = Unspecified<br>All other = Reserved |
| **16** | Event Data 3 | [7:0] – 0xFF |

**Table 86. IPMI Watchdog Sensor Event Trigger Offset – Next Steps**

| Event Trigger Offset | | Description | Next Steps |
|---|---|---|---|
| **00h** | Timer expired, status only | Intel® Server Systems support a BMC watchdog timer, which can check to see whether the OS is still responsive. The timer is disabled by default, and has to be enabled manually. It then requires an IPMI-aware utility in the operating system that resets the timer before it expires. If the timer does expire, the BMC can take action if it is configured to do so (reset, power down, power cycle, or generate a critical interrupt). | If this event is being logged, it is because the BMC has been configured to check the watchdog timer.<br>1. Make sure to have support for this in the OS (typically using a third-party IPMI-aware utility such as ipmitool or ipmiutil along with the OpenIPMI driver).<br>2. If this is the case, it is likely the OS has hung, and investigate OS event logs need to be investigated to determine what may have caused this. |
| **01h** | Hard reset | | |
| **02h** | Power down | | |
| **03h** | Power cycle | | |
| **08h** | Timer interrupt | | |

## 11.2 System Management Interrupt (SMI) Timeout

System management interrupt (SMI) is an interrupt that gets generated so the processor can service server management events (typically memory or PCI errors, or other forms of critical interrupts), in order to log them to the SEL. If this interrupt times out, the system is frozen. The BMC resets the system after logging the event.

**Table 87. SMI Timeout Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | F3h = SMI Timeout |
| **12** | Sensor Number | 06h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 03h (“digital” Discrete) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset = 1h = State Asserted |
| **15** | Event Data 2 | Not used |
| **16** | Event Data 3 | Not used |

### 11.2.1 SMI Timeout – Next Steps

This event normally only occurs after another more critical event.

1. Check the SEL for any critical interrupts, memory errors, bus errors, PCI errors, or any other serious errors.
2. If these are not present, the system locked up before it was able to log the original issue. In this case, low level debug is normally required.

## 11.3 System Event Log Cleared

The BMC logs an SEL clear event. This is only ever the first event in the SEL. The cause of this event is either a manual SEL clear using `selview` or some other IPMI-aware utility, or is done in the factory as one of the last steps in the manufacturing process.

This is an informational event only.

**Table 88. System Event Log Cleared Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 10h = Event Logging Disabled |
| **12** | Sensor Number | 07h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset = 2h = Log area reset/cleared |
| **15** | Event Data 2 | Not used |
| **16** | Event Data 3 | Not used |

## 11.4 System Event Sensor

The BMC supports a System Event sensor and logs an SEL event for a Platform Event Filter (PEF) action, BIOS/Intel® ME out-of-band (OOB) update, and BIOS configuration change from the BMC embedded web server (EWS).

### 11.4.1 System Event – PEF Action

The BMC is configurable to send alerts for events logged into the SEL. These alerts are called Platform Event Filters (PEFs) and are disabled by default. The user must configure and enable this feature. PEF events are logged if the BMC takes action due to a PEF configuration. The BMC event triggering the PEF action is also in the SEL.

This is functionality built into the BMC to allow it to send alerts (SNMP or other) for any event that gets logged to the SEL. PEF filters are turned off by default and have to be enabled manually using a deployment assistant, the syscfg utility, or an IPMI-aware utility.

**Table 89. System Event – PEF Action Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 12h = System Event |
| **12** | Sensor Number | 08h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| **14** | Event Data 1 | [7:6] – 11B = Sensor-specific event extension code in Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset = 4h = PEF Action |
| **15** | Event Data 2 | [7:6] – Reserved<br>[5] – 1b = Diagnostic Interrupt (NMI)<br>[4] – 1b = OEM action<br>[3] – 1b = Power cycle<br>[2] – 1b = Reset<br>[1] – 1b = Power off<br>[0] – 1b = Alert |
| **16** | Event Data 3 | [7:0] – 0xFF |
| **If Event Data 1 [3:0] equals 07h or 08h, Event definition will follow below.** | | |
| **17** | Extended Data 1 | [7:6] - 11b = OEM code in Byte2<br>[5:4] - 00b = unspecified byte 3<br>[3:0] – 07h = image is uploaded<br>[3:0] – 08h = image is lost |
| **18** | Extended Data 2 | [7:0]<br>00h = BIOS Configuration Table<br>01h = BIOS Configuration change<br>02h = BIOS Image<br>03h = ME Image<br>04h = FD Image |
| **19** | Extended Data 3 | [7:0]<br>00h = Firmware Update<br>01h = BIOS Configuration |

This event gets logged if the BMC takes an action due to PEF configuration. Actions can be sending an alert, along with possibly resetting, power cycling, or powering down the system. There is another event that has led to the action; investigate the SEL and PEF settings to identify this event, and troubleshoot accordingly.

### 11.4.2 System Event – OOB Update and BIOS Configuration Change

This event gets logged if the OOB BIOS/ Intel® ME update or BIOS configuration change is triggered from the BMC EWS.

**Table 90. System Event – BIOS/Intel® ME OOB Update And BIOS Configuration Change**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 12h = System Event |
| **12** | Sensor Number | 08h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| **14** | Event Data 1 | [7:6] - 11b = OEM code in Byte2<br>[5:4] - 00b = unspecified byte 3<br>[3:0] – 07h = image is uploaded<br>[3:0] – 08h = image is lost |
| **15** | Event Data 2 | [7:0]<br>00h = BIOS Configuration Table<br>01h = BIOS Configuration change<br>02h = BIOS Image<br>03h = ME Image<br>04h = FD Image |
| **16** | Event Data 3 | [7:0]<br>00h = Firmware Update<br>01h = BIOS Configuration |

## 11.5 BMC Hardware Watchdog Sensor

The BMC supports an IPMI sensor to report that a BMC reset has occurred due to an action taken by the BMC Watchdog feature. A SEL event is logged when either the BMC FW stack is reset or the BMC CPU itself is reset.

**Table 91. BMC Hardware Watchdog Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 28h = Management Subsystem Health |
| **12** | Sensor Number | 0Ah |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 03h (“digital” Discrete) |
| **14** | Event Data 1 | [7:6] - 10b = OEM code in Byte2<br>[5:4] - 10b = OEM code in Byte 3<br>[3:0] – 01h = Event Trigger Offset – State asserted |
| **15** | Event Data 2 | [7:2] – Reserved<br>[1:0] – Threshold Crossed<br>01h = Normal Mode Unstable Boot Threshold<br>02h = Safe Mode Unstable Boot Threshold<br>All other = Reserved |
| **16** | Event Data 3 | [7:0] –Number of successive unstable boot attempts prior to crossing the threshold. |
| **17** | Extended Data 1 | [7] Reserved<br>[6:4] 110b = 6 valid extended data bytes following |

| Byte | Field | Description |
|---|---|---|
| | | [3:2] 00b = Reserved<br>[1:0] 00b = Severity = OK for this sensor (that is, failures should not result in Front Panel status LED to show a warning or fault).<br>General encoding for severity is defined as:<br>00b = OK or non-specified<br>01b = Degraded<br>10b = Warning<br>11b = Failure |
| **18** | Extended Data 2~7 | Firmware revision in use at the time of the fault:<br>Byte 2 = Major rev (binary encoded)<br>Byte 3 = Minor rev (BCD encoded)<br>Byte 4–7 = Build ID (binary) – LSB first |

### 11.5.1 BMC Watchdog Sensor – Next Steps

A SEL event is logged when either the BMC FW stack is reset or the BMC CPU itself is reset.

1. Check the SEL for any other events around the time of the failure.
2. Take note of all IPMI activity that was occurring around the time of the failure. Capture a System BMC Debug Log as soon as possible after the failure. This log can be captured from the Integrated BMC Web Console or by using the syscfg utility (`syscfg/sbmcdl Public filename.zip`). Send the log file to the system manufacturer or Intel representative for failure analysis.

## 11.6 MTM Level Change Sensor

The BMC logs a SEL record when any BMC Manufacturing Test Mode is entered.

**Table 92. MTM Level Change Sensor**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | `28h` = Mgmt Health |
| **12** | Sensor Number | `81h` |
| **13** | Event Direction and Event Type | [7] Event direction<br>`0b` = Assertion Event<br>`1b` = Deassertion Event<br>[6:0] Event Type = `03h` (“digital” Discrete) |
| **14** | Event Data 1 | [7:6] - 10b = OEM code in Byte2<br>[5:4] - 00b = unspecified byte 3<br>[3:0] - 0x01 = Offset from Event/Reading code |
| **15** | Event Data 2 | [7:0] - MTM Level entered |
| **16** | Event Data 3 | [7:0] - 0xFF |

## 11.7 Add-In Module/Board Presence Sensor

Some server baseboards provide dedicated slots for add-in modules/boards (e.g. SAS, IO) or support one or more PCIe* riser boards. For these baseboards, the BMC provides an individual presence sensor to indicate if the module/board is installed.

**Table 93. Add-In Module Presence Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | `15h` = Module/Board |
| **12** | Sensor Number | `0Eh` = OCP Module Presence |

| Byte | Field | Description |
|---|---|---|
| | | 0Fh = SAS Module Presence |
| 13 | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 08h (Digital Discrete) |
| 14 | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset = 01h (Inserted/Present) |
| 15 | Event Data 2 | Not used |
| 16 | Event Data 3 | Not used |

### 11.7.1 Add-In Module Presence – Next Steps

If an unexpected device is removed or inserted, ensure that the module has been seated properly.

## 11.8 BMC Firmware Health Sensor

The BMC tracks the health of each of its IPMI sensors and reports failures by providing a "BMC FW Health" sensor of the IPMI 2.0 sensor type Management Subsystem Health with support for the Sensor Failure offset. Only assertions are logged into the SEL for the Sensor Failure offset. The BMC Firmware Health sensor asserts for any sensor when ten consecutive sensor errors are read. These are not standard sensor events (that is, threshold crossings or discrete assertions). These are BMC Hardware Access Layer (HAL) errors such as $I^2C$ NAKs or internal errors while attempting to read a register. If a successful sensor read is completed, the counter resets to zero.

**Table 94. BMC Firmware Health Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| 11 | Sensor Type | 28h = Management Subsystem Health |
| 12 | Sensor Number | 10h |
| 13 | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| 14 | Event Data 1 | [7:6] – 11b = Sensor-specific event extension code in Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset = 4h = Sensor failure |
| 15 | Event Data 2 | [7:0] - Sensor number that triggered the failure |
| 16 | Event Data 3 | [7:0] – 0xFF |

### 11.8.1 BMC Firmware Health Sensor – Next Steps

1. Check the SEL for any other events around the time of the failure.
2. Take note of all IPMI activity that was occurring around the time of the failure. Capture a System BMC Debug Log as soon as possible after the failure. This log can be captured from the Integrated BMC Web Console or by using the syscfg utility (`syscfg/sbmcdl Public filename.zip`). Send the log file to the system manufacturer or Intel representative for failure analysis.
3. If the failure continues around a specific sensor, replace the board with that sensor.

## 11.9 Firmware Update Status Sensor

The BMC firmware supports a single Firmware Update Status sensor. This sensor is used to generate SEL events related to updates of embedded firmware on the platform. This includes updates to the BMC, BIOS, CPLD and Intel® ME firmware.

This sensor is an event-only sensor that is not readable. Event generation is only enabled for assertion events.

**Table 95. Firmware Update Status Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 2Bh (Version Change) |
| **12** | Sensor Number | 12h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 70h = OEM defined |
| **14** | Event Data 1 | [Bits 7:6] - 10b = OEM code in Byte2<br>[Bits 5:4] - 00b = unspecified Byte3<br>[Bits 3:0] - Offset from Event/Reading code<br>Event Trigger Offset<br>00h = Update started<br>01h = Update completed successfully<br>02h = Update failure |
| **15** | Event Data 2 | [Bits 7:4] Target of update<br>0000b = BMC<br>0001b = BIOS<br>0010b = ME<br>0011b = EWS (Embedded Web server)<br>0101b = CPLD<br>0110b = PCIe* Switch<br>0111b = FD<br>[Bits 3:1] Target instance (zero-based)<br>[Bits 0:0] Reserved |
| **16** | Event Data 3 | Reserved |
| **Note:** All reserved bits must be set to zero. | | |
| **17** | Extended Data 1 | [Bits 7] Reserved<br>[Bits 6:4] = Number of valid extended data bytes following<br>[Bits 3:2] = Reserved<br>[Bits 1:0] Severity = OK for this sensor (that is, failures should not result in Front Panel status LED to show a warning or fault). General encoding for severity is defined as:<br>00b = OK or non-specified<br>01b = Degraded<br>10b = Non-fatal<br>11b = Fatal |
| **18** | Extended Data 2–7 | Firmware revision information (target-specific)<br>If Target of Update is “BMC” then<br>Byte 2 = Major rev (binary encoded)<br>Byte 3 = Minor rev (BCD encoded)<br>Byte 4–7 = Build ID (binary) – LSB first<br>If Target of Update is “BIOS” then<br>Byte 2 = Major rev (BCD encoded)<br>Byte 3 = Minor rev (BCD encoded) |

| Byte | Field | Description |
|---|---|---|
| | | Bytes 4–7 = Build ID (ASCII encoded)<br>If Target of Update is “ME” then<br>Byte 2 = Major rev (binary encoded)<br>Byte 3 = Minor rev1 (BCD encoded)<br>Byte 4 = Minor rev2 (BCD encoded)<br>Byte 5–6 = Build number (BCD Encoded)<br>Byte 7 = Reserved.<br>If Target of Update is “CPLD” then<br>Byte 2 = Major rev (binary encoded) |

## 11.10 Sensor Data Record (SDR) Auto-Config Fault

The BMC provides monitoring of the health status of the sensor data record (SDR) auto-configuration feature. The BMC firmware supports a discrete IPMI sensor for reporting and logging this fault condition.

The single supported sensor offset is asserted when both of the following are true:

- The BMC’s SDR auto-configuration feature fails and
- The BMC’s SDR auto-configuration feature is set to ‘enabled’.

The offset is set to a de-asserted state when either of the following is true:

- The BMC successfully completes an SDR auto-configuration cycle (resulting in auto-detecting the chassis and updating the SDR repository accordingly), regardless of the triggering event or
- The BMC’s SDR auto-configuration feature is set to ‘disable’.

**Table 96. SDR Auto-Config Fault Sensor**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 28h = Mgmt Health |
| **12** | Sensor Number | 87h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 03h (“digital” Discrete) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset = 1h = State Asserted |
| **15** | Event Data 2 | Not used |
| **16** | Event Data 3 | [7:3] – reserved<br>[2:0]: - OEM code for type of auto-config error<br>0h = CFG syntax error<br>1h = Chassis auto-detect error<br>2h = SDR/CFG file mismatch<br>3h = SDR or CFG file corrupted<br>4h = SDR syntax error<br>5h = Parse error<br>All other values reserved |

### 11.10.1 SDR Auto-Config Fault Sensor – Next Steps

1. Check update/upload SDR file and Config file correctly.
2. Check if FRU is valid.
3. Check cable connection especially PSU PMBus cable.

## 11.11 Invalid user name or password sensor

The BMC firmware supports a single Bad User Password sensor. This sensor is used to generate SEL events related to invalid user name or password attempts for BMC access with an IPMI message encapsulated in RMCP/RMCP+. SEL logging will follow the bad password threshold setting from the LAN parameters, and the end user can set/get the bad password threshold by set/getting the LAN configuration command.

**Table 97. Bad User PWD sensor**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 2Ah = Session Audit |
| **12** | Sensor Number | D7h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset<br>02h - Invalid Username or Password<br>03h – Invalid password disable |
| **15** | Event Data 2 | [7:6] Reserved<br>[5:0] User ID<br>0x00 = unspecified |
| **16** | Event Data 3 | [7:4] Reserved<br>[3:0] Channel number |

## 11.12 Remote debug sensor

To reduce the risk of an administrator unknowingly exposing the Remote PECI Debug as a potential attack point, the BMC supports an IPMI sensor to report the configuration state of Remote Debug features

**Table 98. Remote Debug Sensor**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | DBh – OEM Defined. |
| **12** | Sensor Number | DBh |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 7Bh(OEM Defined) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset<br>00h - Remote JTAG Consent<br>01h - Remote JTAG Enabled<br>02h - Remote JTAG Session<br>03h - Remote PECI Enabled<br>04h - Remote PECI Session |
| **15** | Event Data 2 | [7:5] – Reserved<br>[4] – PECI Enabled<br>0b: Disabled<br>1b: Enabled<br>[3] – PECI Session State<br>0b: Idle |

| Byte | Field | Description |
|---|---|---|
| | | 1b: Session in progress<br>[2] – JTAG Debug Consent<br>0b: Consent not given<br>1b: Consent given<br>[1] – JTAG Enabled<br>0b: Disabled<br>1b: Enabled<br>[0] – JTAG Session State<br>0b: Idle<br>1b: Session in progress |
| **16** | Event Data 3 | [7:0] –FFh Unspecified. |
| **17** | Extended Byte1 | [Bit 7] Reserved<br>[Bits 6:4] 110b = 6 valid extended data bytes following<br>[Bits 3:2] 00b = Reserved<br>[Bits 1:0] 00b = Severity = OK for this sensor |
| **18** | Extended Byte 2–7 | Six bytes of last client network connection address. May represent IP address or MAC. |

## 11.13 System Firmware Security Sensor

The BMC supports a Firmware Security Sensor for logging a SEL event for security of BMC Firmware image authentication result in different phase.

**Table 99. System Firmware Security Sensor**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | C3h – OEM defined |
| **12** | Sensor Number | 1Ah |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 71h (OEM Defined) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset<br>00h Authentication Failure of BMC Firmware Image During Boot<br>01h Invalid Security Revision of BMC Firmware Image During Boot<br>02h Authentication Failure of BMC Firmware Image During Update<br>03h Invalid Security Revision of BMC Firmware Image During Update<br>04h Authentication Failure of Signed Region During Update<br>05h Authentication Failure of Signed Region During Boot or Runtime<br>06h Invalid Security Revision of Signed Region During Update<br>07h Invalid Security Revision of Signed Region During Boot or Runtime<br>0Ch Factory Image Booted<br>0Dh Factory Security Revision Downgraded |
| **15** | Event Data 2 | [7:0] - Image Type |
| **16** | Event Data 3 | [7:0] –FFh Unspecified. |

## 11.14 KCS Policy Sensor

KCS (Keyboard Controller Style) interface is the single-wire bus which directly connecting the BMC and the host CPU. Host software including operating system, applications, and BIOS access the BMC through KCS. There is no authentication established for KCS. Communication through this bus is in plain raw data without any authentication and authorization needed. Through KCS, host software can perform operations, which should be in highest privilege. This exposes security risks, as any compromised OS will allow a potential hacker to gain control of the BMC.

To mitigate the security risk, the BMC firmware implements KCS Policy Control Modes to allow an authenticated BMC administrative user to control the level of protection from IPMI commands executed over the KCS channels. Within this generation of BMC firmware, 3 different KCS Policy Control Modes are supported:

**KCS Policy Control Mode – Allow All**

This configuration setting is intended for normal IPMI compliant communications between the Host OS and the BMC. This mode should be used when provisioning the BMC configuration for deployment.

**KCS Policy Control Mode – Deny All**

This configuration setting disables the IPMI KCS command interfaces between the Host OS and the BMC. This is a non-compliant IPMI configuration that will impact the operation of the Server Management Software running on the Host OS. This only applies to the IPMI commands over the KCS interfaces, and does not apply to the authenticated network interfaces to the BMC.

**KCS Policy Control Mode – Restricted**

This configuration setting enables the use of an Access Control List by the BMC Firmware that allows applications executing on the host OS to have access to a limited set of IPMI commands using the KCS interfaces. This is a non-compliant IPMI configuration that may impact the operation of the Server Management Software running on the Host OS. For example, the IPMI commands that are used to provision the BMC configuration settings, or control the state of the Host OS, will be disallowed when the BMC KCS interface is in restricted mode. This only applies to the IPMI commands over the KCS interfaces, and does not apply to the authenticated network interfaces to the BMC.

KCS access is required for the host BIOS during Pre-Boot phase. BIOS communicates with BMC for platform configurations through these channels. Thus BIOS use of KCS is trusted between a platform reset and the end of the BIOS DXE phase. For unconditional and conditional KCS access, two host phases are needed:

1. Pre-Boot phase. This is the phase between platform reset and the end of DXE signaled by the CORE-BIOS-DONE signal before option ROMs are executed. BMC has the below assumptions and KCS handling policies:
   - Core BIOS is trusted
   - KCS commands accepted from BIOS without authentication
   - Transitions to Post-Boot after BIOS sends CORE-BIOS-DONE signal to BMC
2. Post-Boot phase. This is the phase for BIOS to load option ROMs and boot the host OS and host OS execution. BMC assumptions and KCS handling policies are as below:
   - Triggered by CORE-BIOS-DONE signal from BIOS
   - When KCS Policy Control Mode is set to “Allow All”
     - KCS commands accepted from BIOS/EFI, Host OS (Ring0) without authentication
     - Warning Indicator set (EWS banner with explanatory help text, security sensor)
     - Transitions to “Restricted” or “Deny All” with IPMI command from any interface

- When KCS Policy Control Mode is set to “Restricted” or “Deny All”:
  - Triggered by IPMI Command to set KCS Policy Control Mode out of “Allow All”
  - In KCS Restricted Mode, only KCS commands listed in the Access Control List (ACL) are accepted from BIOS/EFI, Host OS (Ring 0) without authentication, or no KCS commands in “Deny All” Mode.
  - May transition back to KCS “Allow All” Mode with:
    - Authenticated IPMI/RMCP+ command from OOB (LAN etc.)
    - Authenticated EWS/Redfish operations
    - Force Firmware Update Jumper (safe mode) allows reset mode from KCS

The following table shows under each KCS Policy Control Mode and host booting phase, KCS commands will be serviced or not.

**Table 100. BMC KCS Policy Control Modes and KCS command service definition**

| KCS Policy Control Modes | Description | Pre-Boot | Post-Boot |
|---|---|---|---|
| **Allow All** | • Default state when customer receives board<br>• Alerts on EWS page and/or security sensor warn owner that the BMC is an unsafe state<br>• Can re-enter KCS Policy Control Mode “Allow All” through authenticated OOB interface or through FFUJ (KCS allowed in safe mode) | All KCS Commands Permitted | All KCS Commands Permitted |
| **Restricted** | • Transitions from “Allow All” when platform owner issues command to set KCS Policy Control Mode (or EWS) | All KCS Commands Permitted | Only commands in the Access Control List are permitted |
| **Deny All** | • Transitions from other modes when platform owner issues commands to set KCS Policy Control Mode (EWS). | All KCS Commands Permitted | No KCS Commands Permitted |

BMC implements OEM type sensor with offsets set to 3 KCS Policy Control Mode. SEL will be captured if KCS Policy Control Mode is changed.

**Table 101. KCS Policy Sensor**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | DAh – KCS Mode |
| **12** | Sensor Number | DAh |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 7Ah(OEM Defined) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset<br>03h = KCS Allow all Mode<br>04h = KCS Restricted mode<br>05h = KCS Deny all mode |
| **15** | Event Data 2 | [7:0] –FFh Unspecified. |
| **16** | Event Data 3 | [7:0] –FFh Unspecified. |

# 12. Hot-Swap Controller Backplane Events

In all Intel® Server Systems product families, the backplanes follow a hybrid architecture, in which the IPMI functionality previously supported in the hot-swap controller (HSC) is integrated into the BMC family.

## 12.1 Hot-Swap Controller (HSC) Backplane Temperature Sensor

There is a thermal sensor on the Hot-Swap Backplane to measure the ambient temperature.

**Table 102. HSC Backplane Temperature Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 01h = Temperature |
| **12** | Sensor Number | 27h = Hot-swap Backplane 1 Temperature (1U HSBP Temp)<br>29h = Hot-swap Backplane 1 Temperature (2U HSBP 1 Temp)<br>2Ah = Hot-swap Backplane 2 Temperature (2U HSBP 2 Temp)<br>2Bh = Hot-swap Backplane 3 Temperature (2U HSBP 3 Temp)<br>E0h = Hot-swap Backplane 4 Temperature (HSBP 4 Temp) |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 01h (Threshold) |
| **14** | Event Data 1 | [7:6] – 01b = Trigger reading in Event Data 2<br>[5:4] – 01b = Trigger threshold in Event Data 3<br>[3:0] – Event Trigger Offset as described in Table 103 |
| **15** | Event Data 2 | Reading that triggered event |
| **16** | Event Data 3 | Threshold value that triggered event |

**Table 103. HSC Backplane Temperature Sensor – Event Trigger Offset – Next Steps**

| Event Trigger | | Assertion Severity | Deassertion Severity | Description | Next Steps |
|---|---|---|---|---|---|
| **00h** | Lower non-critical going low | Degraded | OK | The temperature has dropped below its lower non-critical threshold. | 1. Check for clear and unobstructed airflow into and out of the chassis.<br>2. Ensure the SDR is programmed and correct chassis has been selected.<br>3. Ensure there are no fan failures.<br>4. Ensure the air used to cool the system is within the thermal specifications for the system (typically below 35 °C). |
| **02h** | Lower critical going low | Non-fatal | Degraded | The temperature has dropped below its lower critical threshold. | |
| **07h** | Upper non-critical going high | Degraded | OK | The temperature has gone over its upper non-critical threshold. | |
| **09h** | Upper critical going high | Non-fatal | Degraded | The temperature has gone over its upper critical threshold. | |

## 12.2 Hard Disk Drive Monitoring Sensor

The new backplane design for Intel® Server Systems based on the 3rd Gen Intel® Xeon® Scalable processor family, moves IPMI ownership of the HDD sensors to the BMC. Systems may have multiple storage backplanes. Hard Disk Drive status monitoring is supported through disk status sensors owned by the BMC.

**Table 104. Hard Disk Drive Monitoring Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **11** | Sensor Type | 0Dh = Drive Slot (Bay) |
| **12** | Sensor Number | 60h-68h = Hard Disk Drive 15–23 Status |

| Byte | Field | Description |
|---|---|---|
| | | E2h-E3h = Rear Hard Disk Drive 0–1 Status<br>F0h-FEh = Hard Disk Drive 0–14 Status |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset as described in Table 105 |
| **15** | Event Data 2 | Not used |
| **16** | Event Data 3 | Not used |

### Table 105. Hard Disk Drive Monitoring Sensor – Event Trigger Offset – Next Steps

| Event Trigger | Description | Next Steps |
|---|---|---|
| **00h** | Drive Presence | If during normal operation the state changes unexpectedly, ensure that the drive is seated properly and the drive carrier is properly latched. If that does not work, replace the drive. |
| **01h** | Drive Fault | |
| **07h** | Rebuild/Remap in progress | • This is expected after replacing a hard drive.<br>• This is expected if the system has a hot spare and one of the drives failed. Check logs for which drive has failed.<br>• If this is seen unexpectedly, it could be an indication of a drive that is close to failing. |

# 13. Intel® Management Engine (Intel® ME) Events

The Intel® Management Engine (Intel® ME) controls the platform environmental control interface (PECI) and also contains the Intel® Node Manager (Intel® NM) functionality.

## 13.1 Intel® ME Firmware Health Event

This sensor is used in Platform Event messages to the BMC containing health information including but not limited to firmware upgrade and application errors.

**Table 106. Intel® ME Firmware Health Event Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 002Ch or 602Ch – Intel® ME Firmware |
| **11** | Sensor Type | DCh = OEM |
| **12** | Sensor Number | 17h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 75h (OEM) |
| **14** | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Health event type – 0h (Firmware Status) |
| **15** | Event Data 2 | See Table 107 |
| **16** | Event Data 3 | See Table 107 |

### 13.1.1 Intel® ME Firmware Health Event – Next Steps

In Table 107, Event Data 3 is only noted for specific errors.

If the issue continues to be persistent, provide the content of Event Data 3 to Intel support team for interpretation. Event Data 3 codes are in general not documented, because their meaning only provides some clues, varies, and usually needs to be individually interpreted.

**Table 107. Intel® ME Firmware Health Event Sensor – Next Steps**

| ED2 | ED3 | Description | Next Steps |
|---|---|---|---|
| **00h** | | Recovery GPIO forced. Recovery Image loaded due to recovery MGPIO pin asserted. Pin number is configurable in factory presets. Default recovery pin is MGPIO1. | Deassert MGPIO1 and reset the Intel® ME. |
| **01h** | | Image execution failed. Recovery Image or backup operational image loaded because operational image is corrupted. This may be either caused by flash device corruption or failed upgrade procedure. | Either the flash device must be replaced (if error is persistent) or the upgrade procedure must be started again. |
| **02h** | | Flash erase error. Error during flash erasure procedure. | The flash device must be replaced. |
| **03h** | 00h | Flash state information.<br>Check extended info byte in ED3 whether this is wear-out protection causing this event. If so just wait until wear-out protection expires, otherwise probably the flash device must be replaced (if error is persistent). | Flash partition table, recovery image or factory presets image corrupted. |
| | 01h | | Flash erase limit has been reached. |
| | 02h | | Flash write limit has been reached; writing to flash has been disabled. |
| | 03h | | Writing to the flash has been enabled. |

| ED2 | ED3 | Description | Next Steps |
|---|---|---|---|
| **04h** | | Internal error. Error during firmware execution – FW Watchdog Timeout. | Operational image needs to be updated to other version or hardware board repair is needed (if error is persistent). |
| **05h** | | BMC did not respond to cold reset request and Intel® ME rebooted the platform. | Verify the Intel® Node Manager configuration. |
| **06h** | | Direct Flash update requested by the BIOS. Intel® ME firmware will switch to recovery mode to perform full update from the BIOS. | This is transient state. Intel® ME firmware will return to operational mode after successful image update performed by the BIOS. |
| **07h** | | Manufacturing error. Wrong manufacturing configuration detected by Intel® ME firmware. | The flash device must be replaced (if error is persistent). |
| | 00h | Generic error | |
| | 01h | Wrong or missing VSCC table | |
| | 02h | Wrong sensor scanning period in PIA | |
| | 03h | Wrong device definition in PIA | |
| | 04h | Wrong SMART/CLST configuration | |
| | 05h | Intel® ME FW configuration is inconsistent or out of range | |
| | 0Ah | Percentage of flash write operations, which have been conducted | |
| **08h** | | Persistent storage integrity error. Flash file system error detected. | If error is persistent, restore factory presets using “Force ME Recovery” IPMI command or by doing AC power cycle with Recovery jumper asserted. |
| **09h** | | Firmware Exception. | Restore factory presets using “Force ME Recovery” IPMI command or by doing AC power cycle with Recovery jumper asserted. If this does not clear the issue, reflash the SPI flash. |
| **0Ah** | | Flash Wear-Out Protection Warning. Warning threshold for number of flash operations has been exceeded. | No immediate repair action needed. This is just a warning event. |
| **0Ch** | | Invalid SCMP interface state. | This is a warning event that indicates that the SCMP interface was in a wrong state. The interface shall normally recover from such a state. It likely indicates an SW or FW defect. SW or FW fix is needed in such a situation. |
| **0Dh** | | PECI over DMI interface error. This is a notification that PECI over DMI interface failure was detected and it is not functional anymore. It may indicate the situation when PECI over DMI was not configured by BIOS or a defect, which may require a CPU Host reset to recover from. | ME FW will switch to Serial PECI interface when connected directly to chipset. Recovery via CPU Host reset or platform reset. |
| **0Eh** | | MCTP interface error. This is a notification that MCTP interface failure was detected and it is not functional anymore. It may indicate the situation when MCTP was not configured by BIOS or a defect, which may need a Host reset to recover from. | Recovery via CPU Host reset or platform reset. |
| **0Fh-FFh** | | Reserved. | |

## 13.2 Intel® Node Manager Exception Event

An Intel® Node Manager exception event is sent each time maintained policy power limit is exceeded over Correction Time Limit.

**Table 108. Intel® Node Manager Exception Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 002Ch or 602Ch – Intel® ME Firmware |
| **11** | Sensor Type | DCh = OEM |
| **12** | Sensor Number | 18h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 72h (OEM) |
| **14** | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3] – Node Manager Policy event<br>0 – Reserved<br>1 – Policy Correction Time Exceeded – Policy did not meet the contract for the defined policy. The policy continues to limit the power or shut down the platform based on the defined policy action.<br>[2] – Reserved<br>[1:0] – 00b |
| **15** | Event Data 2 | [4:7] – Reserved<br>[0:3] – Domain Id<br>00h – Entire platform.<br>01h – CPU subsystem.<br>02h – Memory subsystem.<br>03h – HW Protection.<br>04h – High Power I/O subsystem.<br>Others – Reserved |
| **16** | Event Data 3 | Policy Id |

### 13.2.1 Intel® Node Manager Exception Event – Next Steps

This is an informational event. Next steps depend on the policy that was set. See the *Intel® Node Manager Specification* for more details.

## 13.3 Intel® Node Manager Health Event

An Intel® Node Manager Health Event message provides a runtime error indication about Intel® Intelligent Power Node Manager's health. Types of service that can send an error are defined as follows:

- Misconfigured policy Error reading power data.
- Error reading inlet temperature.

**Table 109. Intel® Node Manager Health Event Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 002Ch or 602Ch – Intel® ME Firmware |
| **11** | Sensor Type | DCh = OEM |
| **12** | Sensor Number | 19h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 73h (OEM) |
| **14** | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Health Event Type = 02h (Sensor Node Manager) |
| **15** | Event Data 2 | [7:4] – Error type<br>0–7 – Reserved<br>8 – Outlet Temperature Reading Failure<br>9 – Volumetric Airflow Reading Failure<br>10 – Policy Misconfiguration<br>11 – Power Sensor Reading Failure<br>12 – Inlet Temperature Reading Failure<br>13 – Host Communication error<br>14 – Real-time clock synchronization failure<br>15 – Platform shutdown initiated by NM policy due to execution of action defined by Policy Exception Action<br>[3:0] – Domain Id<br>00h – Entire platform<br>01h – CPU subsystem<br>02h – Memory subsystem<br>03h – HW Protection<br>04h – High Power I/O subsystem<br>Others – Reserved |
| **16** | Event Data 3 | If Error type = 10 or 15 <Policy Id><br>If Error type = 11 <Power Sensor Address><br>If Error type = 12 <Inlet Sensor Address><br>Otherwise set to 0. |

### 13.3.1 Intel® Node Manager Health Event – Next Steps

Misconfigured policy can happen if the max/min power consumption of the platform exceeds the values in policy due to hardware reconfiguration.

First occurrence of not acknowledged event will be retransmitted no faster than every 300 milliseconds.

Real-time clock synchronization failure alert is sent when Intel® NM is enabled and capable of limiting power, but within 10 minutes the firmware cannot obtain valid calendar time from the host side, so Intel® NM cannot handle suspension periods.

Next steps depend on the policy that was set. See the *Intel® Node Manager Specification* for more details.

## 13.4 Intel® Node Manager Operational Capabilities Change

This message provides a runtime error indication about Intel® Intelligent Power Node Manager's operational capabilities. This applies to all domains.

Assertion and deassertion of these events are supported.

**Table 110. Intel® Node Manager Operational Capabilities Change Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 002Ch or 602Ch – Intel® ME Firmware |
| **11** | Sensor Type | DCh = OEM |
| **12** | Sensor Number | 1Ah |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 74h (OEM) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Current state of Operational Capabilities. Bit pattern:<br>0 – Policy interface capability<br>0 – Not Available<br>1 – Available<br>1 – Monitoring capability<br>0 – Not Available<br>1 – Available<br>2 – Power limiting capability<br>0 – Not Available<br>1 – Available |
| **15** | Event Data 2 | Not used |
| **16** | Event Data 3 | Not used |

### 13.4.1 Intel® Node Manager Operational Capabilities Change – Next Steps

Policy Interface available indicates that Intel® Intelligent Power Node Manager is able to respond to the external interface about querying and setting Intel® Intelligent Power Node Manager policies. This is generally available as soon as the microcontroller is initialized.

Monitoring Interface available indicates that Intel® Intelligent Power Node Manager has the capability to monitor power and temperature. This is generally available when firmware is operational.

Power limiting interface available indicates that Intel® Intelligent Power Node Manager can do power limiting and is indicative of an Advanced Configuration and Power Interface (ACPI)-compliant OS loaded (unless the OEM has indicated support for non-ACPI compliant OS).

Current value of not acknowledged capability sensor is retransmitted no faster than every 300 milliseconds.

Next steps depend on the policy that was set. See the *Intel® Node Manager Specification* for more details.

## 13.5 Intel® Node Manager Alert Threshold Exceeded

Policy Correction Time Exceeded Event is sent each time maintained policy power limit is exceeded over Correction Time Limit.

**Table 111. Intel® Node Manager Alert Threshold Exceeded Sensor Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 002Ch or 602Ch – Intel® ME Firmware |
| **11** | Sensor Type | DCh = OEM |
| **12** | Sensor Number | 1Bh |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 72h (OEM) |
| **14** | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3] = Node Manager Policy event<br>0 – Threshold exceeded<br>1 – Policy Correction Time Exceeded – Policy did not meet the contract for the defined policy. The policy continues to limit the power or shut down the platform based on the defined policy action.<br>[2] – Reserved<br>[1:0] – Threshold Number.<br>0 to 2 – Threshold index |
| **15** | Event Data 2 | [7:4] – Reserved<br>[3:0] – Domain Id<br>00h – Entire platform<br>01h – CPU subsystem<br>02h – Memory subsystem<br>03h – HW Protection<br>04h – High Power I/O subsystem<br>Others – Reserved |
| **16** | Event Data 3 | Policy ID |

### 13.5.1 Intel® Node Manager Alert Threshold Exceeded – Next Steps

First occurrence of not acknowledged event will be retransmitted no faster than every 300 milliseconds.

First occurrence of Threshold exceeded event assertion/deassertion will be retransmitted no faster than every 300 milliseconds.

Next steps depend on the policy that was set. See the *Node Manager Specification* for more details.

## 13.6 Intel® Node Manager SmaRT and CLST Sensor

The Intel® Node Manager monitors the power supplies in the system for an SMBALERT# assertion in the power supplies. If it deems that this is caused by what would be considered a system SmaRT/CLST event, then it throttles the total system power to prevent the system failure. This is logged as a SmaRT/CLST event.

### Table 112. Intel® Node Manager Smart/CLST Event Typical Characteristics

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 002Ch or 602Ch – Intel® ME Firmware |
| **11** | Sensor Type | DCh = OEM |
| **12** | Sensor Number | B2h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 03h (Digital Discrete) |
| **14** | Event Data 1 | [7:6] – 01b = Previous state and/or severity in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Offset from Event/Reading Code<br>0h – State Deasserted<br>1h – State Asserted |
| **15** | Event Data 2 | [7:4] – Optional offset from 'Severity' Event/Reading Code.<br>00h - transition to OK<br>01h - transition to noncritical from OK<br>02h - transition to critical from less severe<br>03h - transition to unrecoverable from less severe<br>04h - transition to noncritical from more severe<br>05h - transition to critical from unrecoverable<br>06h - transition to unrecoverable<br>07h - monitor<br>08h - informational<br>[3:0] - Fh |
| **16** | Event Data 3 | Corresponding Power Supply Status sensor number or 0 if the source of SmaRT&CLST assertion is external (for example BMC). |

### Table 113. Smart And CLST Sensor Severity Codes

| Severity Code | Description | PSU Condition |
|---|---|---|
| **00h** | Transition to OK | All present PSU faults disappeared. |
| **01h** | Transition to noncritical from OK | SMBAlert# has been asserted by PSU but it should be ignored (e.g. PSU goes to off state due insufficient input voltage) |
| **02h** | Transition to critical from less severe | SMBAlert# was asserted due to one of the following PSU events:<br>• UV_Fault or<br>• OT_Warning or<br>• OC_Warning |
| **04h** | Transition to noncritical from more severe | There has been a critical condition and one of the following events happened:<br>• UV Fault lasted more than preconfigured time (default 500ms) or<br>• PSU causing disappeared or<br>• PSU unit become off or<br>• OT warning lasted more than 500ms for one PSU and there is another PSU present with no OT warning. |

### 13.6.1 Intel® Node Manager SmaRT/CLST Event – Next Steps

1. Look for any other events in the System Event Log for correlation.
2. Ensure that there are no issues with any of the installed power supplies.
3. If there is only one power supply then consider adding an additional power supply to handle the required load.

# 14. Microsoft Windows* Records

With Microsoft Windows Server* 2003 R2 and later versions, an Intelligent Platform Management Interface (IPMI) driver was added. This added the capability of logging some operating system events to the SEL. The driver can write multiple records to the SEL for the following events:

- Boot-up
- Shutdown
- Bug Check / Blue Screen

## 14.1 Boot Up Event Records

When the system boots into the Microsoft Windows* OS, two events can be logged. The first is a boot-up record and the second is an OEM event. These are informational only records.

**Table 114. Boot Up Event Record Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 0041h – System Software with an ID = 20h |
| **11** | Sensor Type | 1Fh = OS Boot |
| **12** | Sensor Number | 00h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset = 1h = C: boot completed |
| **15** | Event Data 2 | Not used |
| **16** | Event Data 3 | Not used |

**Table 115. Boot Up OEM Event Record Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **1, 2** | Record ID | ID used for SEL Record access. |
| **3** | Record Type | [7:0] – DCh = OEM timestamped, bytes 8–16 OEM defined. |
| **4–7** | Timestamp | Time when the event was logged. The least significant byte is first. |
| **8–10** | IPMI Manufacturer ID | 0137h (311d) = IANA enterprise number for Microsoft. |
| **11** | Record ID | Sequential number reflecting the order in which the records are read. The numbers start at one for the first entry in the SEL and continue sequentially to n, the number of entries in the SEL. |
| **12–15** | Boot Time | Timestamp of when the system booted into the OS. |
| **16** | Reserved | 00h |

## 14.2 Shutdown Event Records

When the system shuts down from the Microsoft Windows* OS, multiple events can be logged. The first is an OS Stop/Shutdown Event Record; this can be followed by a shutdown reason code OEM record, and then zero or more shutdown comment OEM records. These are all informational only records.

**Table 116. Shutdown Reason Code Event Record Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 0041h – System Software with an ID = 20h |
| **11** | Sensor Type | 20h = OS Stop/Shutdown |
| **12** | Sensor Number | 00h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset = 3h = OS Graceful Shutdown |
| **15** | Event Data 2 | Not used |
| **16** | Event Data 3 | Not used |

**Table 117. Shutdown Reason OEM Event Record Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| **1, 2** | Record ID | ID used for SEL Record access |
| **3** | Record Type | [7:0] – DDh = OEM timestamped, bytes 8–16 OEM defined |
| **4–7** | Timestamp | Time when the event was logged. LS byte first. |
| **8–10** | IPMI Manufacturer ID | 0137h (311d) = IANA enterprise number for Microsoft |
| **11** | Record ID | Sequential number reflecting the order in which the records are read. The numbers start at one for the first entry in the SEL and continue sequentially to n, the number of entries in the SEL. |
| **12–15** | Shutdown Reason | Shutdown Reason code from the registry (LSB first):<br>HKLM/Software/Microsoft/Windows/CurrentVersion/Reliability/shutdown/ReasonCode |
| **16** | Reserved | 00h |

**Table 118. Shutdown Comment OEM Event Record Typical Characteristics**

| Byte | Field | Description |
|---|---|---|
| 1, 2 | Record ID | ID used for SEL Record access |
| 3 | Record Type | [7:0] – DDh = OEM timestamped, bytes 8–16 OEM defined |
| 4–7 | Timestamp | Time when the event was logged. The least significant byte is first. |
| 8–10 | IPMI Manufacturer ID | 0137h (311d) = IANA enterprise number for Microsoft<br>0157h (343d) = IANA enterprise number for Intel<br>The value logged depends on the Intelligent Management Bus Driver (IMBDRV) that is loaded. |
| 11 | Record ID | Sequential number reflecting the order in which the records are read. The numbers start at one for the first entry in the SEL and continue sequentially to n, the number of entries in the SEL. |
| 12–15 | Shutdown Comment | Shutdown Comment from the registry (least significant bit first):<br>HKLM/Software/Microsoft/Windows/CurrentVersion/Reliability/shutdown/Comment |
| 16 | Reserved | 00h |

## 14.3 Bug Check/Blue Screen Event Records

When the system experiences a bug check (blue screen), multiple records are written to the event log. The first record could be a Bug Check, Blue Screen OS Stop, or Shutdown Event Record followed by multiple Bug

Check/Blue Screen code OEM records that contain the Bug Check/Blue Screen codes. This information can be used to determine what caused the failure.

### Table 119. Bug Check/Blue Screen – OS Stop Event Record Typical Characteristics

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 0041h – System Software with an ID = 20h |
| **11** | Sensor Type | 20h = OS Stop/Shutdown |
| **12** | Sensor Number | 00h |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| **14** | Event Data 1 | [7:6] – 00b = Unspecified Event Data 2<br>[5:4] – 00b = Unspecified Event Data 3<br>[3:0] – Event Trigger Offset = 1h = Runtime Critical Stop (that is, “core dump”, “blue screen”) |
| **15** | Event Data 2 | Not used |
| **16** | Event Data 3 | Not used |

### Table 120. Bug Check/Blue Screen Code OEM Event Record Typical Characteristics

| Byte | Field | Description |
|---|---|---|
| **1, 2** | Record ID | ID used for SEL Record access |
| **3** | Record Type | [7:0] – DEh = OEM timestamped, bytes 8–16 OEM defined |
| **4–7** | Timestamp | Time when the event was logged. The least significant byte is first. |
| **8–10** | IPMI Manufacturer ID | 0137h (311d) = IANA enterprise number for Microsoft<br>0157h (343d) = IANA enterprise number for Intel<br>The value logged depends on the Intelligent Management Bus Driver (IMBDRV) that is loaded. |
| **11** | Sequence Number | Sequential number reflecting the order in which the records are read. The numbers start at 1 for the first entry in the SEL and continue sequentially to n, the number of entries in the SEL. |
| **12–15** | Bug Check/Blue Screen Data | The first record of this type contains the Bug Check/Blue Screen Stop code and is followed by the four Bug Check/Blue Screen parameters. The east significant byte is first.<br>Each of the Bug Check/Blue Screen parameters requires two records each.<br>Both of the two records for each parameter have the same Record ID.<br>There is a total of nine records. |
| **16** | Operating system type | 00 = 32-bit OS<br>01 = 64-bit OS |

# 15. Linux* Kernel Panic Records

The Open IPMI driver supports the ability to put semi-custom and custom events in the system event log if a panic occurs. Enable the “Generate a panic event to all BMCs on a panic” option to get one event on a panic in a standard IPMI event format. Enable the “Generate OEM events containing the panic string” option to get a set of OEM events holding the panic string.

**Table 121. Linux* Kernel Panic Event Record Characteristics**

| Byte | Field | Description |
|---|---|---|
| **8, 9** | Generator ID | 0021h – Kernel |
| **10** | EvM Rev | 03h = IPMI 1.0 format |
| **11** | Sensor Type | 20h = OS Stop/Shutdown |
| **12** | Sensor Number | The first byte of the panic string (0 if no panic string) |
| **13** | Event Direction and Event Type | [7] Event direction<br>0b = Assertion Event<br>1b = Deassertion Event<br>[6:0] Event Type = 6Fh (Sensor Specific) |
| **14** | Event Data 1 | [7:6] – 10b = OEM code in Event Data 2<br>[5:4] – 10b = OEM code in Event Data 3<br>[3:0] – Event Trigger Offset = 1h = Runtime Critical Stop (a.k.a. “core dump”, “blue screen”) |
| **15** | Event Data 2 | The second byte of the panic string |
| **16** | Event Data 3 | The third byte of the panic string |

**Table 122. Linux* Kernel Panic String Extended Record Characteristics**

| Byte | Field | Description |
|---|---|---|
| **1, 2** | Record ID | ID used for SEL Record access |
| **3** | Record Type | [7:0] – F0h = OEM non-timestamped, bytes 4–16 OEM defined |
| **4** | Slave Address | The slave address of the card saving the panic |
| **5** | Sequence Number | A sequence number (starting at zero) |
| **6–16** | Kernel Panic Data | These hold the panic sting. If the panic string is longer than 11 bytes, multiple messages are sent with increasing sequence numbers. |

# Appendix A. Glossary

| Term | Definition |
|---|---|
| **ACPI** | Advanced Configuration and Power Interface |
| **AER** | Advanced Error Reporting |
| **BIOS** | Basic Input/Output System |
| **BMC** | Baseboard Management Controller |
| **CATERR** | Catastrophic Error |
| **CFM** | Cubic Feet per Minute |
| **CLST** | Closed Loop System Throttling |
| **DIMM** | Dual Inline Memory Module |
| **DLLP** | Data Link Layer Packet |
| **DTS** | Digital Thermal Sensor |
| **ECC** | Error Correction Code |
| **EWS** | Embedded Web Server |
| **FP** | Front Panel |
| **FRU** | Field Replaceable Unity |
| **GPGPU** | General-purpose computing on graphics processing unit |
| **HDD** | Hard Disk Drive |
| **HSC** | Hot-Swap Controller |
| **HSBP** | Hot-Swap Backplane |
| **IANA** | Internet Assigned Numbers Authority |
| **IERR** | Internal Error |
| **IPMI** | Intelligent Platform Management Interface |
| **IPMB** | Intelligent Platform Management Bus |
| **LUN** | Logical Unit Number |
| **MCTP** | Management Component Transport Protocol |
| **Intel® ME** | Intel® Management Engine |
| **MSR** | Model Specific Register |
| **NIC** | Network Interface Controller |
| **Intel® NM** | Intel® Node Manager |
| **NMI** | Non-Maskable Interrupt |
| **OEM** | Original Equipment Manufacturer |
| **OOB** | Out-of-band |
| **OS** | Operating System |
| **PCH** | Platform Controller Hub |
| **PCIe*** | PCI Express* |
| **PECI** | Platform Environmental Control Interface |
| **PEF** | Platform Event Filters |
| **PERR** | Parity Error |
| **POST** | Power On Self Test |
| **PSU** | Power Supply Unit |
| **PWM** | Pulse Width Modulation |
| **Intel® QPI** | Intel® QuickPath Interconnect |
| **RAS** | Reliability, Availability, Serviceability |
| **RqSA** | Requester's Slave Address |

| Term | Definition |
|---|---|
| **SAS** | Serial Attached SCSI |
| **SATA** | Serial ATA |
| **SCMP** | Software Configuration Management Plan |
| **SDR** | Sensor Data Record |
| **SmaRT** | Smart Ride Through |
| **SEL** | System Event Log |
| **SERR** | System Error |
| **SMI** | System Management Interrupt |
| **SSB** | South Side Bridge |
| **TLP** | Transaction Layer Packet |
| **TPS** | Technical Product Specification |
| **TSOD** | Temperature Sensor on DIMM |
| **VR** | Voltage Regulator |
| **VSCC** | Vendor Specific Component Capabilities |