SR870BN4 Error Reference Guide

Revision 1.0

October 2003

Enterprise Platforms & Services Marketing

Revision History

Date	Revision Number	Modifications
01/2002	0.5	Initial Release.
04/2002	0.6	Update Machine Check Error Handling section, update SEL data tables.
10/2003	1.0	Updated sensor and beep code tables.

Disclaimers

Information in this document is provided in connection with Intel® products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications. Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

This document contains information on products in the design phase of development. Do not finalize a design with this information. Revised information will be published when the product is available. Verify with your local sales office that you have the latest datasheet before finalizing a design.

The SR870BN4 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, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Table of Contents

1. Introduction 1
1.1 Document Organization 1
1.2 SEL Overview 1
2. EFI-Based SELViewer Utility 2
3. SR870BN4 SEL Data Tables 3
3.1 SR870BN4 Generator ID Codes 3
3.2 SR870BN4 Sensor Codes 3
4. SR870BN4 Machine Check Error Handling 7
4.1 Classification of Errors 7
4.1.1 Error Types 7
4.1.2 Error Signaling 7
4.2 Error Reporting 8
4.3 Thresholding 9
4.4 SEL Event Log Format for Machine Check Errors 9
5. SR870BN4 PCI Device IDs 12
6. BIOS POST Error Codes and Messages 13
7. POST Codes 18
7.1 North and South Port 80/81 Cards 18
7.2 POST Codes 18
7.2.1 POST Codes Module Map 18
7.2.2 Specific POST code Modules 20
7.3 Recovery Port 80 Codes 30
8. Beep Codes 31
8.1.1 Memory Test Failure 31
8.1.2 Recovery Beep Codes 32
8.1.3 BMC Beep Code Generation 32
Appendix A: Glossary I
Appendix B: Reference Documents IV
Appendix C: Index V

List of Figures

Figure 1. SEL Viewer Utility 2

List of Tables

Table 1. SAL 3.0 MCA Records 8
Table 2. SEL Event Logs for Machine Check Errors 10
Table 3: Onboard PCI Devices and Slots 12
Table 4. Error Code Classification 13
Table 5. General POST Code Module Numbers for Itanium™ – Based Platforms 19
Table 6. SAL-A POST Codes (BSP Only) 20
Table 7. SAL-B POST Codes 21
Table 8. SAL-F POST Codes 24
Table 9. IA-32 POST Codes 25
Table 10. EFI POST Codes 28
Table 11. ACPI POST Codes 29
Table 12. SAL Runtime POST Codes 29
Table 13. Recovery POST Codes 30
Table 14. Error Beep Codes 31
Table 15. POST Memory Beep Error Codes – Debug Port Encoding List 31
Table 16. Recovery Mode Beep Codes 32
Table 17: BMC Beep Codes 32

1. Introduction

This document is an error reference guide for the SR870BN4 server system.

1.1 Document Organization

Section 1: An introduction to the SEL.

Section 2: A brief introduction to the EFI-based SEL Viewer utility.

Section 3: SEL Data Tables.

Section 4: MCA Error Handling including SEL event format for machine check events.

Section 5: SR870BN4 PCI Device IDs

Section 6: BIOS POST error codes and messages. These error codes and messages appear on the video and are also logged in the system event log.

Section 7: SR870BN4 BIOS POST codes. A port 80/81 card is required to view these codes.

Section 8: A list of beep codes generated by the SR870BN4 platform.

Appendix A: Glossary.

Appendix B: Reference Documents.

Appendix C: Index.

1.2 SEL Overview

The System Event Log (SEL) is a non-volatile repository for event messages. Event messages contain information about system events and anomalies that occur on the server. They can be triggered by BIOS, event generators, or sensors. Some event messages are the result of normal happenings, such as a normal server boot, or possible minor problems like a disconnected keyboard. Other events may indicate internal failures such as a component overheat condition where thresholds, or ranges of acceptable values, exist. As with other system events, if at any time a component crosses one of these defined thresholds, an event message will be generated.

Regardless of the event, the appropriate management controller generates an event message. Event messages are passed to the Baseboard Management Controller (BMC), the primary management controller on Intel® server systems. The BMC passes the event message to the SEL where it becomes available for querying by an SEL Viewer utility.

The SEL Viewer provides an interface for the server administrator to view information in the SEL. The SEL Viewer is available through the Intel Server Management (ISM) or the EFI-based SEL Viewer utility. The system administrator can use this information to monitor the server for warnings and potential critical problems.

2. EFI-Based SELViewer Utility

The EFI-based SEL Viewer utility is used to view the SEL records from Itanium™ -based servers.

The SEL Viewer provides support for the user to perform the following:

- Examine all SEL entries stored in the non-volatile storage area of the server in text form or in hexadecimal.
- Examine previously stored SEL entries from a file in text form or in hexadecimal.
- Save the SEL entries to a file.
- Clear the SEL entries from the non-volatile storage area.
- Sort the SEL records by various fields such as timestamp, sensor type number, event description, and generator ID.

Five columns of SEL data can be viewed from the EFI SEL Viewer Utility:

1. Number of Event.
2. Time Stamp.
3. Sensor Type and Number.
4. Event Description.
5. Generator ID.

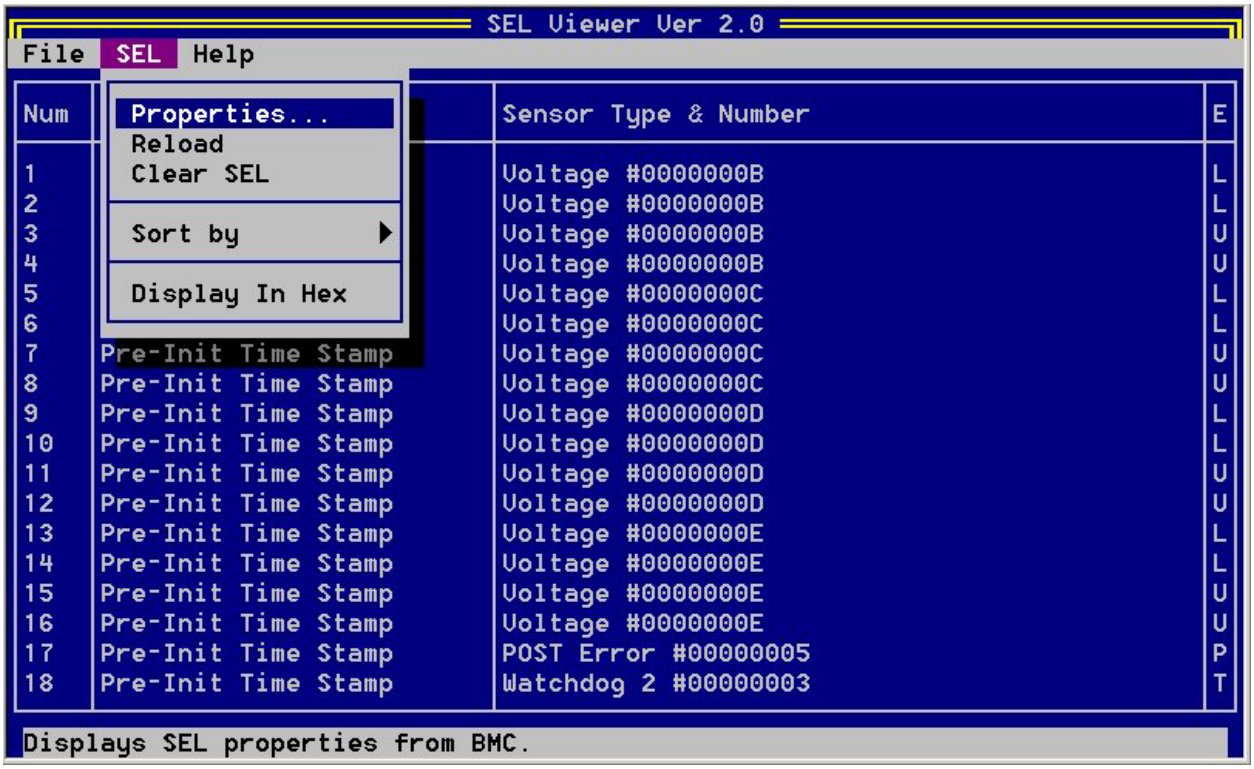

Figure 1. SEL Viewer Utility

3. SR870BN4 SEL Data Tables

The tables in this section provide information on the data provided by the SEL Viewer utility.

3.1 SR870BN4 Generator ID Codes

Generator ID	Description
20 00	BMC
C0 00	HSC
0x31 00 -0x3F 00	System BIOS or System SW

3.2 SR870BN4 Sensor Codes

Sensor Type	Sensor Number	Sensor Name
01		**Temperature**
	01h	SCSI BP Temperature
	30h	Memory Board 1 Temp
	31h	Memory Board 2 Temp
	32h	IO Board Temp 1
	33h	IO Board SIOH Temp
	34h	IO Board Temp 3
	35h	CPU Board Amb Temp
	36h	CPU Board SNC Temp
	98h	Proc 1 Temp
	99h	Proc 2 Temp
	9Ah	Proc 3 Temp
	9Bh	Proc 4 Temp
02		**Voltage**
	10h	IO Board +1.3V
	11h	IO Board +1.5V
	12h	IO Board +1.8V
	15h	IO Board +3.3V SB
	16h	IO Board +3.3V A
	17h	IO Board +3.3V B
	18h	IO Board V_{BAT}
	19h	IO Board +5V A
	1Ah	IO Board +5V B
	1Bh	IO Board +12V
	1Ch	IO Board -12V
	1Dh	CPU Board +1.2V
	1Eh	Memory Board 1 +1.25V
	1Fh	Memory Board 2 +1.25V
	20h	CPU Board +1.3V
	21h	CPU Board +1.5V

Sensor Type	Sensor Number	Sensor Name
	22h	CPU Board +1.8V
	23h	CPU Board +3.3V SB
	24h	CPU Board +12V SB
	25h	IO Riser Board +12V SB
	26h	IO Riser Board +2.5V
	27h	IO Riser Board +1.5V SB
	28h	IO Riser Board +1.5V
	29h	IO Riser Board +5V SB
	2Ah	I/O Board SCSI +5V
	60h	LVDS SCSI channel 1 terminator 1
	61h	LVDS SCSI channel 1 terminator 2
	62h	LVDS SCSI channel 1 terminator 3
	63h	LVDS SCSI channel 2 terminator 1
	64h	LVDS SCSI channel 2 terminator 2
	65h	LVDS SCSI channel 2 terminator 3
	7Eh	CPU Board 12V SB Power Good
	7Fh	Node Power Good
	A4h	Proc 1 Power Pod Good
	A5h	Proc 2 Power Pod Good
	A6h	Proc 3 Power Pod Good
	A7h	Proc 4 Power Pod Good
04		**Fan**
	40h	Tach Fan 1
	41h	Tach Fan 2
	42h	Tach Fan 3
	43h	Tach Fan 4
	82h	Fan 1 Present
	83h	Fan 2 Present
	84h	Fan 3 Present
	85h	Fan 4 Present
06		**Security Violation Attempt**
	04h	Platform Security Violation
07		**Processor**
	90h	Proc 1 Status
	91h	Proc 2 Status
	92h	Proc 3 Status
	93h	Proc 4 Status
08		**Power Supply**
	70h	Power Supply 1
	71h	Power Supply 2
	74h	I/O Board 3.3V D2D 3
	75h	I/O Board 3.3V D2D 2
	76h	I/O Board 5V D2D 1

Sensor Type	Sensor Number	Sensor Name
	77h	I/O Board 5V D2D 2
	78h	Processor Board 3.3V D2D 1
	79h	Processor Board 2.5V D2D 1
	7Ah	Processor Board 2.5V D2D 2
	7Bh	Memory Board 1 1.25V D2D
	7Ch	Memory Board 2 1.25V D2D
09		**Power Unit**
	01h	Power Unit Status
	02h	Power Unit Redundancy
0D		**Hot Swap Drive Sensors**
	02h	Hot Swap Drive 1 Status
	03h	Hot Swap Drive 2 Status
	04h	Hot Swap Drive 3 Status
	09h	Hot Swap Drive 1 Present
	0Ah	Hot Swap Drive 2 Present
	0Bh	Hot Swap Drive 3 Present
0F		**System Firmware Progress**
	06h	POST Error
10		**Event Logging**
	09h	Event Logging Disabled
13		**Critical Interrupt**
	07h	FP Diag Interrupt (Front Panel Diag Interrupt)
15		**Module / Board**
	80h	Memory Board 1 Present
	81h	Memory Board 2 Present
	86h	IO Board Present
	87h	IO Riser Board Present
	88h	CPU Board Present
	89h	SCSI Board Present
21		**Slot / Connector**
	E0h	PHP Slot 1
	E1h	PHP Slot 2
	E2h	PHP Slot 3
	E3h	PHP Slot 4
	E4h	PHP Slot 5
	E5h	PHP Slot 6
	E6h	PHP Slot 7
	E7h	PHP Slot 8

Sensor Type	Sensor Number	Sensor Name
23		**Watchdog**
	03h	BMC Watchdog2
C7		**OEM**
	50h	Fan Boost Memory Board 1 Temp
	51h	Fan Boost Memory Board 2 Temp
	52h	Fan Boost IO Board Temp 1
	53h	Fan Boost IO Board SIOH Temp
	54h	Fan Boost IO Board Temp 3
	55h	Fan Boost CPU Board Ambient Temp
	56h	Fan Boost CPU Board SNC Temp
	A0h	Fan Boost Proc 1 Temp
	A1h	Fan Boost Proc 2 Temp
	A2h	Fan Boost Proc 3 Temp
	A3h	Fan Boost Proc 4 Temp

4. SR870BN4 Machine Check Error Handling

This section gives an overview of the implementation of machine check error handling on the SR870BN4 server system. For additional details about Itanium-based system error generation and error handling, refer to the *Itanium™ Processor Family Error Handling Guide* (document number: 249278-002) and the *Itanium™ System Abstraction Layer Specification* (document number: 245359-005). Both documents can be downloaded from the web at developer.intel.com.

4.1 Classification of Errors

Error events are classified by the processor and platform into three basic groups. This section provides a summary of the different error types and signaling methods defined by the IPF Machine Check architecture and implemented in the SR870BN4 platform.

4.1.1 Error Types

- **Fatal:** A fatal error is an error where the state has been corrupted, and the error may or may not be contained. The platform will signal a fatal error when the integrity of the platform or subsystem cannot be determined. These errors cannot be corrected by hardware, firmware, or system software, and a reset of the system or subsystem is required.

- **Recoverable/Uncorrectable:** An error has been detected that cannot be corrected by hardware or firmware. However, the operating integrity of platform hardware and system state has been maintained. These errors may or may not be recoverable (determined by system software capabilities).

- **Correctable:** An error has been detected and corrected by hardware, or by processor/platform firmware.

4.1.2 Error Signaling

- **Corrected Machine Check Interrupt (CMCI):** Corrected processor errors are signaled as a CMCI to system software. For example, L1 tag parity errors on shared lines or thermal events are corrected by the processor (logic or the PAL). System software must insure that the interrupt handler for CMCI executes on the same processor that signaled the corrected error event.

- **Corrected Platform Event Interrupt (CPEI):** These interrupts are signaled by the platform or the SAL. These include errors that are corrected by the platform (such as single-bit ECC error in memory) and errors that are not correctable by the platform. In either case, the error is contained (i.e., data poisoning), and the platform can still function reliably. One example of an uncorrected error is a 2XECC error detected on a write to memory.

- **Machine Check Events:** A processor machine check occurs when the processor detects a fatal or recoverable error during execution of instructions or when the processor is signaled by the platform to enter machine check.

There are two types of machine check events: local and global. A local MCA is when an individual processor enters machine check. Some examples of local machine checks include a Distributed Translation Lookaside Buffer (DTLB) data parity error, or when the processor consumes data with an uncorrectable error.

A machine check is global when all processors enter machine check. On the SR870BN4 platform, the method used to get all processors into machine check are the BINIT# and BERR# signals. When a processor takes a local machine check, it may escalate the error to a global machine check to transition other processors to a known state and/or for error containment. For example, the processor may assert BINIT# in response to a transaction time-out event.

The SR870BN4 platform does not assert BINIT#, only BERR#. BERR# is asserted for platform fatal errors and when an uncorrectable error is detected on outbound data.

For more information on the SR870BN4 implementation of machine check error handling, refer to the S*R870BN4 SAL Error Handling Specification*.

4.2 Error Reporting

SR870BN4 machine check error handling allows enhanced error reporting of processor and platform errors. These errors are prioritized and signaled to system hardware and software. System software (PAL/SAL) provides well-defined APIs for application software to acquire information about system errors in the form of standard data structures. These errors are logged to non-volatile storage and/or made available for consumption by application software during runtime. These errors are in the MCA records and they are based on the *Itanium*™ *System Abstraction Layer Specification* Rev 3.0.

On the SR870BN4, based on the MCA records, system events related to Field Replaceable Units (FRUs) are logged in the BMC SEL. Each MCA record results in the generation of one or more corresponding BMC SEL events. In addition, an auxiliary log entry event will be logged corresponding to each MCA record. The SEL messages are IPMI 1.5-compliant platform event messages.

The following rules are applied to the translation of SAL 3.0 MCA records to IPMI 1.5-compliant platform event messages :

Table 1. SAL 3.0 MCA Records

MCA SAL Record Section Type	SEL Event : Sensor Type	SEL event: Event Data Bytes
Processor	Processor IERR	SMBIOS Type 4 0-based index Error Severity
PCI Bus PERR/SERR	Critical Interrupt PERR SERR	PCI Bus number
PCI Bus Other Errors	Critical Interrupt Bus Correctable error Bus Uncorrectable error	None

PCI Component	Critical Interrupt PERR SERR	PCI Bus, Device, Function info
Memory Device	Memory Error Correctable Uncorrectable	SMBIOS Type 16 0-based index SMBIOS Type 17 0-based index
Other	Critical Interrupt Bus Correctable error Bus Uncorrectable error	

4.3 Thresholding

MCA classifies errors into one of three categories: corrected, recoverable, and fatal. In general, corrected errors will not affect the operation of the sytem and therefore may occur repeatedly (fatal and most recoverable errors result in a system reset.) In some cases, such as a stuck bit in a memory DIMM, a corrected error may occur with a very high frequency. In this scenario, the system may experience a performance degredation due to excessive amounts of time spent in the error logging routines. In addition, the BMC SEL has a finite size and may be quickly filled with duplicate errors. To help alleviate this problems, a thresholding agorithm has been applied to the BMC SEL logging routines. If the threshold is crossed, a special “event disabled” SEL entry will be created and the the BMC SEL logging code will not attempt to send future platform event message commands for that error type to the BMC. This greatly reduces the amount of time spent in the SEL logging routines and avoids overrunning the BMC SEL log storage. This thresholding in no way affects the ability of the OS to receive notification and service CPEIs or CMCIs, nor does it disable any error correction logic in the chipset. Any disabled event reporting will be re-enabled on the next reboot.

Corrected errors are grouped into four categories: Processor, Memory, PCI PERR, and Generic Bus. History for each category is maintained separately. Recoverable and fatal errors are not thresholded, only corrected errors. On the SR870BN4, the maximum number of errors that can occur for each category is “10” within one hour. If this threshold is crossed, a special ‘**Event Logging Disabled’** SEL entry will be logged.

4.4 SEL Event Log Format for Machine Check Errors

The following table shows the machine check errors that will be logged for the SR870BN4, and the corresponding SEL Event Log format. For details on System Management BIOS (SMBIOS) Type 4, Type 16 and 17, refer to the *System Management BIOS Reference Specificaton* available on www.dmtf.org.

Table 2. SEL Event Logs for Machine Check Errors

Error Type	Gen ID	EvMRev	Sen Type	Sen #	Ev Dir/Type	Data 1	Data 2	Data 3
Processor Specific								
Fatal	0x31	0x4	0x7	N/A	0x6F	0xA0	Index to SMBIOS Type4 record	Severity - 0x01
Uncorrectable	0x31	0x4	0x7	N/A	0x6F	0xA0	Index to SMBIOS Type4 record	Severity - 0x00
Correctable	0x31	0x4	0x7	N/A	0x6F	0xA0	Index to SMBIOS Type4 record	Severity - 0x02
Memory DIMM specific								
Uncorrectable	0x33	0x4	0xC	N/A	0x6F	0x81	0xFF	Bit 7:6 – Index to SMBIOS Type16 record Bit 5:0 – Index to SMBIOS Type 17 record
Correctable	0x33	0x4	0xC	N/A	0x6F	0x80	0xFF	Bit 7:6 – Index to SMBIOS Type16 record Bit 5:0 – Index to SMBIOS Type 17 record
PCI Device Specific								
SERR	0x31	0x4	0x13		0x6F	0xA4	PCI Bus #	Bit 7:3 -DEV# Bit 2:0 -Func#
PERR	0x31	0x4	0x13		0x6F	0xA5	PCI Bus #	Bit 7:3 -DEV# Bit 2:0 -Func#
PCI Bus PERR/SERR								
SERR	0x31	0x4	0x13	N\A	0x6F	0x84	PCI Bus #	0xFF
PERR	0x31	0x4	0x13	N\A	0x6F	0x85	PCI Bus #	0xFF
Processor Bus, LPC Bus, SP port, HL Bus, non-specific Bus Errors								
Uncorrectable	0x31	0x4	0x13	N/A	0x6F	0x08	0xFF	0xFF
Correctable	0x31	0x4	0x13	N/A	0x6F	0x07	0xFF	0xFF

Event Logging Disabled (Thresholding)								
SBE Memory Logging Disabled	0x31	0x4	0x10	N/A	0x6F	0x00	0xFF	0xFF
Bus Correctable Logging Disabled	0x31	0x4	0x10	N/A	0x6F	0xF1	0x13	0x26
Proc Correctable Logging Disabled	0x31	0x4	0x10	N/A	0x6F	0xF1	0x07	0x20
PCI PERR Logging Disabled	0x31	0x4	0x10	N/A	0x6F	0xF1	0x13	0x24
System Event (MCA Event Indicator)								
Aux Log Entry	0x31	0x4	0x12	N/A	0x6F	0xC3	0x20	0xFF
Aux Log Entry	0x31	0x4	0x12	N/A	0x6F	0xC3	0x00	0xFF

5. SR870BN4 PCI Device IDs

The SR870BN4 server has the following PCI devices and slots on the I/O board:

Table 3: Onboard PCI Devices and Slots

Device Description	PCI Bus	Bus Number	Device ID	Function Number
SNC	FSB	0xFF	0x18	0,1.2
SIOH	SNC	0xFF	0x1C	
MRH-D	SNC	0xFF	0x018	1
ICH4	Internal	0x0	30	0
LPC		0x0	31	0
IDE Controller		0x0	31	1
USB Controller1		0x0	31	2
USB Controller2		0x0	31	4
Video	Internal	0x0		
NIC	Internal	0x1	08	0
SCSI Controller	A (embedded)	Dynamic		
PCI slot 1 (bus shared with embedded SCSI)	A (100Mhz half-size)	Dynamic		
PCI slot 2	B (100Mhz half-size)	Dynamic		
PCI slot 3	B (100Mhz half-size)	Dynamic		
PCI slot 4 (PCI/HL)	C (100Mhz half-size)	Dynamic		
PCI slot 5	C (100Mhz full-size)	Dynamic		
PCI slot 6	D (133Mhz full-size)	Dynamic		
PCI slot 7	E (133Mhz full-size)	Dynamic		
PCI slot 8	F (133Mhz full-size)	Dynamic		

6. BIOS POST Error Codes and Messages

The following error codes are relevant to the SR870BN4 server. The system BIOS displays POST error messages on the video screen and are also logged in the SEL. The SR870BN4 BIOS will prompt the user to press a key in case of serious errors.

Error Code Classification

Red: Critical events that require user interaction. BIOS POST will pause with a message requesting to Press F1, F2, or ESC. This error code type is indicated in the table below as a YES in the column heading *Pause On Boot.*

Yellow: Non critical events. BIOS POST will continue after a brief pause and does not require user interaction. This error code type is indicated in the table below as a NO in the column heading *Pause On Boot.*

Table 4. Error Code Classification

Error Code	Error Message	Pause on Boot	Recommended User Action
103	CMOS Battery Failure	Yes	Replace the battery.
105	CMOS Checksum Failure	Yes	Clear CMOS.
107	Insert Key Pressed	Yes	
0109	Keyboard Stuck Key	Yes; user input required	May appear after reboot if there was a keyboard error, such as not being plugged in. Could also be a timing problem with the keyboard.
011B	Date/Time Not Set	Yes; user input required	Enter BIOS setup and set the date and time.
0120	CMOS clear	Yes; user input required	Move the CMOS switch to the inactive position.
0121	Password clear	Yes; user input required	Move the Password switch to the inactive position.
0140	PCI Error	Yes	Remove add-in cards individually to see where error stops. If necessary, replace I/O board.
0141	PCI Memory Allocation Error	Yes	Remove add-in cards individually to see where error stops. If necessary, replace I/O board.
0142	PCI IO Allocation Error	Yes	Remove add-in cards individually to see where error stops. If necessary, replace I/O board.
0143	PCI IRQ Allocation Error	Yes	Remove add-in cards individually to see where error stops. If necessary, replace I/O board.
0144	Shadow of PCI ROM Failed	Yes	Remove add-in cards individually to see where error stops. If necessary, replace I/O board.

Error Code	Error Message	Pause on Boot	Recommended User Action
0145	PCI ROM not found	Yes	It can mean valid option ROM bar present, but no actual physical option ROM present. Informative, no user action required.
0146	Insufficient Memory to Shadow PCI ROM	Yes	This is due to lack of option ROM space in the BIOS. This error can be resolved by disabling all of the option ROMS on all devices except for the boot device.
8100	Processor 01 failed BIST	Yes; user input required	Replace Processor 01.
8101	Processor 02 failed BIST	Yes; user input required	Replace Processor 02.
8102	Processor 03 failed BIST	Yes; user input required	Replace Processor 03.
8103	Processor 04 failed BIST	Yes; user input required	Replace Processor 04.
8110	Processor 01 Internal error(IERR)	Yes; user input required	Processor signal , read by BMC and BIOS gets the information from BMC. Indicates hardware failure. User should replace processor.
8111	Processor 02 Internal error(IERR)	Yes; user input required	Processor signal read by BMC and BIOS gets the information from BMC. Indicates hardware failure. User should replace processor.
8112	Processor 03 Internal error(IERR)	Yes; user input required	Processor signal read by BMC and BIOS gets the information from BMC. Indicates, hardware failure. User should replace processor.
8113	Processor 04 Internal error(IERR)	Yes; user input required	Processor signal read by BMC and BIOS gets the information from BMC. Indicates hardware failure. User should replace processor.
8130	Processor 01: Disabled	Yes; user input required	Retest processor. If error persists, Replace processor 01.
8131	Processor 02: Disabled	Yes; user input required	Retest processor. If error persists, Replace processor 02.
8132	Processor 03: Disabled	Yes; user input required	Retest processor. If error persists, Replace processor 03.
8133	Processor 04: Disabled	Yes; user input required	Retest processor. If error persists, Replace processor 04.
8140	Processor 01: failed FRB level 3 timer	Yes; user input required	Retest processor. If error persists, Replace processor 01.
8141	Processor 02: failed FRB level 3 timer	Yes; user input required	Retest processor. If error persists, Replace processor 02.
8142	Processor 03: failed FRB level 3 timer	Yes; user input required	Retest processor. If error persists, Replace processor 03.
8143	Processor 04: failed FRB level 3 timer	Yes; user input required	Retest processor. If error persists, Replace processor 04.
8150	Processor 01: failed initialization on last boot	Yes; user input required	Retest processor. If error persists, hardware failure. User should replace processor.

Error Code	Error Message	Pause on Boot	Recommended User Action
8151	Processor 02: failed initialization on last boot	Yes; user input required	Retest processor. If error persists, hardware failure. User should replace processor.
8152	Processor 03: failed initialization on last boot	Yes; user input required	Retest processor. If error persists, hardware failure. User should replace processor.
8153	Processor 04: failed initialization on last boot	Yes; user input required	Retest processor. If error persists, hardware failure. User should replace processor.
8180	BIOS does not support current stepping for Processor 01	Yes	User should replace processor with a supported stepping.
8181	BIOS does not support current stepping for Processor 02	Yes	User should replace processor with a supported stepping.
8182	BIOS does not support current stepping for Processor 03	Yes	User should replace processor with a supported stepping.
8183	BIOS does not support current stepping for Processor 04	Yes	User should replace processor with a supported stepping.
8193	CPUID, Processor Steppings are different	Yes	Use processors with the same stepping (e.g. C0, C1). User information, typically n and n -1 stepping mixes allowed.
8194	CPUID, Processor Families are different	Yes; user input required	Use processors that are the same speed. CPUID has 3 fields=>stepping, family and model. If any of the fields mismatch, then user would get this message. User has to replace processor with the same kind.
8196	Processor Models are Different	Yes; user input required	Use processors that are the same speed and stepping.
8197	Processor Speed mismatch	Yes	Use processors that are the same speed and stepping.
8300	Baseboard Management Controller failed to function	Yes	Try to reflash the BMC with the most current version that is compatible with your system configuration. If problem persists, may require board replacement.
84F3	Baseboard Management Controller in Update Mode	Yes; user input required	The BMC recovery jumper is in the active position and the BMC can be reflashed. After reflashing, move the jumper to the inactive position.
84FF	System Event Log Full	Yes	Clear the SEL in BIOS setup.
8500	Multi-bit Error Detected Row1. Row 1 mapped out	Yes	Verify the affected memory and replace with correct memory.
8501	Multi-bit Error Detected Row2. Row 2 mapped out	Yes	Verify the affected memory and replace with correct memory.
8502	Multi-bit Error Detected Row3. Row 3 mapped out	Yes	Verify the affected memory and replace with correct memory.
8503	Multi-bit Error Detected Row4. Row 4 mapped out	Yes	Verify the affected memory and replace with correct memory.

Error Code	Error Message	Pause on Boot	Recommended User Action
8504	Persistent Single-bit Error Detected Row1. Row 1 mapped out	Yes	Verify the affected memory and replace with correct memory.
8505	Persistent Single-bit Error Detected Row2. Row 2 mapped out	Yes	Verify the affected memory and replace with correct memory.
8506	Persistent Single-bit Error Detected Row3. Row 3 mapped out	Yes	Verify the affected memory and replace with correct memory.
8507	Persistent Single-bit Error Detected Row4. Row 4 mapped out	Yes	Verify the affected memory and replace with correct memory.
8508	Memory Mismatch detected Row1. Row 1 mapped out	Yes	Verify the affected memory and replace with like memory.
8509	Memory Mismatch detected Row2. Row 2 mapped out	Yes	Verify the affected memory and replace with like memory.
850A	Memory Mismatch detected Row3. Row 3 mapped out	Yes	Verify the affected memory and replace with like memory.
850B	Memory Mismatch detected Row4. Row 4 mapped out	Yes	Verify the affected memory and replace with like memory.
850C	DIMM1, memory board 1 defective.	Yes	Verify the affected memory and replace with good memory.
850D	DIMM2, memory board 1 defective.	Yes	Verify the affected memory and replace with good memory.
850E	DIMM3, memory board 1 defective.	Yes	Verify the affected memory and replace with good memory.
850F	DIMM4, memory board 1 defective.	Yes	Verify the affected memory and replace with good memory.
8510	DIMM5, memory board 1 defective.	Yes	Verify the affected memory and replace with good memory.
8511	DIMM6, memory board 1 defective.	Yes	Verify the affected memory and replace with good memory.
8512	DIMM7, memory board 1 defective.	Yes	Verify the affected memory and replace with good memory.
8513	DIMM8, memory board 1 defective.	Yes	Verify the affected memory and replace with good memory.
8514	DIMM1, memory board 2 defective.	Yes	Verify the affected memory and replace with good memory.
8515	DIMM2, memory board 2 defective.	Yes	Verify the affected memory and replace with good memory.
8516	DIMM3, memory board 2 defective.	Yes	Verify the affected memory and replace with good memory.
8517	DIMM4, memory board 2 defective.	Yes	Verify the affected memory and replace with good memory.
8518	DIMM5, memory board 2 defective.	Yes	Verify the affected memory and replace with good memory.
8519	DIMM6, memory board 2 defective.	Yes	Verify the affected memory and replace with good memory.
851A	DIMM7, memory board 2 defective.	Yes	Verify the affected memory and replace with good memory.
851B	DIMM8, memory board 2 defective.	Yes	Verify the affected memory and replace with good memory.

7. POST Codes

In order to indicate progress through BIOS POST, and in special cases where errors are encountered during BIOS POST, there are three common mechanisms which shall be employed by the SR870BN4 BIOS. The first method is to display port 80/81 codes to a I^2C* adapter connected to the processor baseboard. The second common method is the use of beep codes, encoded beep sequences emitted by the PC speaker when an error is encountered. Beep codes are employed only before the display screen is enabled, and generally indicate *fatal* errors. Beep codes are coupled with special port 80 error codes. The final method is to display an error message to the display screen.

7.1 North and South Port 80/81 Cards

In the case of the SR870BN4 server, this port 80 card is a custom device attached to I^2C ports in two different places on the server. One port 80 device serves the north flash ROM and the other serves the south flash ROM.

7.2 POST Codes

7.2.1 POST Codes Module Map

The SR870BN4 server employs a novel post code scheme. Post codes assigned make use of the fact that the SR870BN4 utilizes port 80h and 81h. This gives the SR870BN4 16 bits to encode. The following rules apply to the post code encoding:

Bit 15: 1 – IA64 code being executed, 0 – IA-32 code being executed

Bit 14: 1 – system stopped due to known failure, 0 – progress indication

All other module bits remain unmodified.

Bit 13: 1 – fault or trap (no change in module numbers), 0 – normal execution

In case of fault or trap, only bit 13 is set and other bits are left on modified. This allows us to detect which module produces the fault.

Bit 12: Reserved

Bit 11-4: Module type

Bit 3-0: Sub module type

The module number and sub module number are a in 4-bit boundary to allow us to decode quickly by the numbers. The module number identifies the major module such as memory, PCI, ACPI, etc. The sub module identifies the sub function such as SPD read in progress, ECC error, and DIMM mismatch for memory module.

Module names and numbers are listed in the following tables.

Secret Decoder:

Bit 11:8 – 0xF stack-less code being executed, 0xD-0x0 – memory is available

Table 5. General POST Code Module Numbers for Itanium™ – Based Platforms

Code Value (bit 8 = 1, bits 11:4 shown below)				**Module**	**Display**
0xFF				Reserved	North
0xFE				Reset Condition	North
0xFD				Node BSP selection	North
0xFC				Early node init (SNCPEIM)	North
0xFB				Processor health/setup (CVDR PEIM)	North
0xFA				PAL/FW health status	North
0xF9-F7				Memory Initialization	North
SUB MODULES				Memory Initialization	North
BITS					
15:12	11:8	7:4	3:0		
8	F	7	0	Pass1 Entry	
8	F	7	1	RAC Initialization (Mem_DoRacInitialization())	
8	F	7	2	Validate DIMMs (Mem_ValidateInstalledConfiguration())	
8	F	7	3	Program MIRs/MITs (Mem_DoMirMitProgram())	
8	F	7	4	Calculate CAS (Mem_CalcSysCas())	
C	F	7	4	Calculate CAS Error Loop	
8	F	7	5	Program CAS (Mem_SetMrhdCasLatency())	
8	F	7	6	Set Mrhd DIMM Geomentry (Mem_SetMrhdDimmGeometry())	
8	F	7	7	Perform SLEW rate calibration (Mem_DoSlewRateCalibration)	
8	F	7	8	Mem_InitDimmAndSetCasLatencyAndBurst()	
8	F	7	9	DDR delay Calibration (Mem_DoDdrDelayCalibration())	
8	F	8	0	DIMM path latency Calibration	
8	F	8	1	DIMM Strobe Delay Calibration	
8	F	8	2	Configure SNC timing	
8	F	8	3	Set timings for write pattern	
8	F	9	0	Levelization	
8	F	9	8	Reconfigure memory	
C	F	9	F	Levelization failed. No Memory Found	
0xF6				Memory Test	North
0xF5				Platform Discovery	North
0xF4-F3				SBSP selection & Platform Init	North

<table>
<tr><th colspan="4">Code Value (bit 8 = 1, bits 11:4 shown below)</th><th>Module</th><th>Display</th></tr>
<tr><td colspan="4">0xF2</td><td>Memory Autoscan (stackless)</td><td>North</td></tr>
<tr><td colspan="4">SUB MODULES</td><td rowspan="2">Memory Autoscan</td><td rowspan="5">North</td></tr>
<tr><td colspan="4">BITS</td></tr>
<tr><td>15:12</td><td>11:8</td><td>7:4</td><td>3:0</td><td></td></tr>
<tr><td>8</td><td>F</td><td>2</td><td>0</td><td>Pass1 Entry</td></tr>
<tr><td>8</td><td>F</td><td>2</td><td>1</td><td>Process Auto Scan Input</td></tr>
<tr><td>8</td><td>F</td><td>2</td><td>2</td><td>Execute Auto scan (C- code)</td></tr>
<tr><td>8</td><td>F</td><td>2</td><td>3</td><td>Process Auto Scan Output</td></tr>
<tr><td colspan="4">0xF1</td><td>Recovery stackless</td><td>North</td></tr>
<tr><td colspan="4">0xF0</td><td>Reserved</td><td>North</td></tr>
<tr><td colspan="4">0xEF-0xEE</td><td>Memory Autoscan C-code</td><td>North</td></tr>
<tr><td colspan="4">0xED-E8</td><td>Recovery C-Code</td><td></td></tr>
<tr><td colspan="4">0xE8-0xE6</td><td>HOB</td><td>North</td></tr>
<tr><td colspan="4">0xE5-0xC1</td><td>Reserved</td><td>North</td></tr>
<tr><td colspan="4">0xC0</td><td>SALA to SALB/DXE handoff</td><td>North</td></tr>
<tr><td colspan="4">0xB0-0xBF</td><td>Reserved</td><td>North</td></tr>
<tr><td colspan="4">0xAF-80</td><td>Reserved for SAL
MCA, INIT, PMI</td><td>North</td></tr>
<tr><td colspan="4">0x7F to 0x60</td><td>SAL-B codes
SAL-B
SAL_C
SAL_F</td><td>
South
South
South</td></tr>
<tr><td colspan="4">0x60</td><td>SAL to EFI handoff</td><td>South</td></tr>
<tr><td colspan="4">0x5F to 0x50</td><td>EFI</td><td>South</td></tr>
<tr><td colspan="4">0x4F to 0x40</td><td>ACPI</td><td>South</td></tr>
</table>

7.2.2 Specific POST code Modules

7.2.2.1 SAL-A Module

The SAL-A POST codes are defined in the following table.

Table 6. SAL-A POST Codes (BSP Only)

Code Number	Meaning	Display
0x8FE0	Reset Condition	North
0x8FD0	Node BSP selection	North
0x8FC0	Early node init (SNCPEIM)	North
0x8FB0	Processor health/setup (CVDR PEIM)	North
0x8FA0	PAL/FW health status	North
0x8F70	Memory Initialization Entry	North
0x8F71	RAC Initialization (Mem_DoRacInitialization())	North

0x8F72	Validate DIMMs (Mem_ValidateInstalledConfiguration())	North
0x8F73	Program MIRs/MITs (Mem_DoMirMitProgram())	North
0x8F74	Calculate CAS (Mem_CalcSysCas())	North
0xCF74	Calculate CAS Error Loop	North
0x8F75	Program CAS (Mem_SetMrhdCasLatency())	North
0x8F76	Set Mrhd DIMM Geomentry (Mem_SetMrhdDimmGeometry())	North
0x8F77	Perform SLEW rate calibration (Mem_DoSlewRateCalibration)	North
0x8F78	Mem_InitDimmAndSetCasLatencyAndBurst()	North
0x8F79	DDR delay Calibration (Mem_DoDdrDelayCalibration())	North
0x8F80	DIMM path latency Calibration	North
0x8F81	DIMM Strobe Delay Calibration	North
0x8F82	Configure SNC timing	North
0x8F83	Set timings for write pattern	North
0x8F90	Levelization	North
0x8F98	Reconfigure memory	North
0xCF9F	Levelization failed. No Memory Found	North
0x8F60	Memory Test	North
0x8F50	Platform Discovery	North
0x8F40	SBSP selection	North
0x8F50	Platform Init	North
0x8F20	Memory Autoscan entry	North
0x8F21	Process Auto Scan Input	North
0x8F22	Process Auto Scan Output	North
0x8F10	Recovery code entry	North
0x8ED0	Recovery C-Code Entry	North
0x8ED1	Recovery Reading media	North
0xCEDF	Recovery Reading error	North
0x8EC0	Recovery program start	North
0x8EC1	Recovery program success	North
0xCECF	Recovery programming error	North
0x8E80	PEIM Handoff block entry	North
0x8C00	SALA to SALB/DXE handoff	North
0x8AF0 to 0x8800	Reserved for MCA, INIT, PMI	North

7.2.2.2 SAL-B Module

Table 7. SAL-B POST Codes

Code Number	BSP, APs, Both	Meaning	Display
0x87FF	BSP+APs	First check point. Initialize cr.iva/ar.eflag/ar.cflg/cr.lrr0/cr.lrr1/cr.ifa/cr.itir	South
0x87FE	BSP only	Initialize io_base address, CPU#, health, etc. for CPU's. Initialize	South

	And BSP+APs	min_state_area for all CPU's (cpu_data_base+cpu_bspstore_base+cpu_health)cpu_data_base points to min state save area. TOM below and above 4G. Allocate sal_mp_info_table data and sal_efi stack area and legacy_stack (temp). Initialize legacy stack top and bottom for temporary use during POST only. INT_15,(FN# F788 in EM code) uses INT-8 timer tick for frequency calculation. (BSP+APs) Save ID,EID, Initialize BSPSTORE,SP.	
0x87FD	BSP only	Search FIT for legacy BIOS.	South
0x07FD	BSP only	Then hang, if not found. If found copy top 64K legacy boot block ROM at xxxx:0000.	South
0x87FC	BSP only	Search for legacy_nvm module (sal_legacy_nvm_module_1d).	South
0x07FC	BSP only	Then hang, if not found. Else continue by saving in RAM.	South
0x87FB	BSP only	Search for efi_nvm module (sal_efi_nvm_module_1e).	South
0x07FB	BSP only	Then hang, if not found. Else continue by saving in RAM. Reserve 128k memory for NVM emulation.	South
0x87FA	BSP only	Search for acpi_dsdt module (sal_acpi_data_module_16) Ask for Address, size, type.	South
0x07FA	BSP only	Then hang, if not found. Else continue by saving in RAM.	South
0x87F9	BSP only	Search for addition information acpi_dsdt module Ask for size, align, and scratch buff size.	South
0x07F9	BSP only	Then hang, if not found. Else continue by saving in RAM.	South
0x87F8	BSP only	Search for addition information acpi_dsdt module. Initialize scratch buffer.	South
0x07F8	BSP only	Then hang, if not found. Else continue by saving in RAM.	South
0x87F7	BSP only	Reserve ACPI_64 and ACPI_32 data area. Reserve MP table data area. Save SAL data base & size. SAL shadow top (PELoader + SAL_F).	South
0x87F6	BSP only	Cache flush after PELoader shadow.	South
0x07F6	BSP only	Hang, on ERROR.	South
0x87F5	BSP only	Search for information on SAL_F module (sal_f_module_12). By size, align, and scratch buff size.	South
0x07F5	BSP only	Then hang, if not found or Information ERROR. SAL shadow bottom (PELoader + SAL_F) Find SAL_F page size. Align to next 32K boundary and save address and size.	South
0x87F4	BSP only	Search for addition information SAL_F module Initialize scratch buffer.	South
0x07F4	BSP only	Then hang, if not found. Else continue by saving in RAM.	South
0x87F3	BSP only	Cache flush after SAL shadowed.	South

0x07F3	BSP only	Hang on ERROR.	South
*0x87F2	BSP only	Initialize sal data top address Physical equals to virtual for runtime use and above 4G Load Call backs for byte/word checkpoint display entry and Address. SAL PMI address EFI to SAL call back address SAL procedure address SAL SST base and address SAL procudure entry base inside SST Buildtime address where SAL_PROC entry is stored Buildtime GP Runtime GP SAL SST size.	South
0x87F1	BSP only	Load PAL module.	South
0x87F0	BSP+APs	BSP Shadow PAL module, initialize PAL shadow base, size, proc ptr initialize PAL procedure address entry & checksum AP's PAL PMI base will be set.	South
0x07F0	BSP+APs	Hang on ERROR.	South
0x87EF	BSP only	Cache flush after PAL shadow.	South
0x07EF	BSP only	Hang on ERROR.	South
0x87EE	BSP only	Find PAL shadow size + align through SAL call.	South
0x07EE	BSP only	Hang on ERROR.	South
0x87ED	BSP only	Find # of CPU's present in the system, # of CPU, # of IOAPIC.	South
0x07ED	BSP only	Hang on ERROR.	South
0x87EC	BSP only	Search for addition information EFI module (sal_efi_module_15) size, align, and scratch buff size. Initialize scratch buffer.	South
0x07EC	BSP only	Hang if ERROR.	South
0x87EB	BSP only	Save maximum (PAL,EFI) shadow size and alignment. Save PAL(ia32)/EFI shadow top address, size, alignment. EFI module shadow base address (virtual/Physical), size, bottom address (DATA+SAL+PAL+EFI). Update virtual address entries in translation register descriptor, addresses in MDT,	South
0x87EA	BSP+APs	Cache flush shadow.	South
0x07EA	BSP + APs	Hang on ERROR.	South
0x87E9	BSP + APs	PAL call for memory Test for SELF TEST(pal_mem_for_test_25).	South
0x07E9	BSP + APs	Hang, if Memory ERROR.	South
0x87E8	BSP + APs	PAL call for PAL test (pal_test_proc_102) and save results.	South
0x07E8	BSP + APs	Hang, if late self test ERROR NOTE: this can be skipped by a build switch.	South
0x87E7	BSP + APs	PAL Call for pal_bus_get_features function # (pal_bus_get_features_09).	South
0x07E7	BSP + APs	Hang if ERROR.	South
0x87E6	BSP + APs	Set buslock mask=1 (non-atomic) By PAL Call PAL Bus Set Feature (pal_bus_set_features_0a).	South
0x07E6	BSP + APs	Hang if ERROR.	South

0x87E5	BSP + APs	Set PMI entry point PAL Call (pal_pmi_entrypoint_20).	South
0x07E5	BSP + APs	Hang if ERROR.	South
0x87E4	BSP + APs	PAL Cache Summary by PAL Call (pal_cache_summary_04).	South
0x07E4	BSP + APs	Hang if ERROR	South
0x87E3	BSP + APs	PAL Cache Information set. PAL Call cache_info_02.	South
0x07E3	BSP + APs	Hang, if ERROR.	South
0x87E2	BSP + APs	pal_mc_register_mem_1b/find CPU min state pointer. Should be able now to initialize health,bsp/ap,cache size line size, sapic ver, and cpuid. Set minimal state save area, BSPSTORE and SP.	South
0x87E1	BSP + APs	Cache flush shadow.	South
0x07E1	BSP + APs	Hang if ERROR.	South
0x87E0	BSP + APs	Program IVA,ITR(0) for PAL,SAL runtime code & data area cr.iva/cr.ifa/cr.itir/itr[r0].	South
0x87DF	BSP + APs	Clear semaphore and wait for all CPUs to synchronize.	South
0x87DE	BSP + APs	Sort CPU health. Already sorted for 2nd level BSP selection. Store BSP/AP flag for respective CPU.	South
0x87DD	APs	Setup for interrupt wakeup reinitialization of BSPSTORE and SP if needed. Wait for interrupt wakeup.	South
0x87DC	BSP only	Switch to virtual address Control register programming SET in PSR bn(44), it(36), rt(27), dt(17), ic(13) . Clear task priority register=cr.tpr. Clear interruption function state register-cr.ifs. Set legacy BIOS cs.base and ss.base. Set es,ds,fs,gs=0 with 4G limit Legacy BIOS module (eip). Give control at xxxx:e05b to IA-32 code.	South

7.2.2.3 SAL-F Module

Table 8. SAL-F POST Codes

Code Value	BSP, APs, Both	Meaning	Display
0x87BF	BSP	First check point. Check point in v6b00_83_ip2x. Update EBDA entry inside SST Create EFI memory descriptor Update SST checksum.	South
0x87BE	BSP	Check point near v6b00_83_5 Search FIT for ACPI module (SAL_C_module_17) and get size, align, scratch buff size.	South
0x07BE	BSP	Hang if ERROR.	South
0x87BD	BSP	Load image by module type (sal_c_module_17). Use PELoader.	South
0x07BD	BSP	Hang if not found. Get entry point, and GP value.	South
0x87BC	BSP	Load image by module type (sal_c_module_17). Flush cache.	South
0x07BC	BSP	Hang on ERROR. Build MP & ACPI table.	South
0x87BB	BSP	Initialize memory manager(0x0) by call to SAL_C.	South

0x07BB	BSP	Hang on ERROR.	South
0x87BA	BSP	Feed system information (0x1) with call to SAL_C.	South
0x07BA	BSP	Hang on ERROR.	South
0x87B9	BSP	Initialize MP table v1.4 (0x2) with call to SAL_C	South
0x07B9	BSP	Hang on ERROR.	South
0x87B8	BSP	Initialize IA-32 ACPI v1.1 (0x3) with call to SAL_C	South
0x07B8	BSP	Hang on ERROR.	South
0x87B7	BSP	Initialize IA64 ACPI v1.1 (0x4) with call to SAL_C	South
0x07B7	BSP	Hang on ERROR.	South
0x87B6	BSP	Initialize IA-32&IA64 ACPI v2.0 (0x5) with call to SAL_C	South
0x07B6	BSP	Hang on ERROR.	South
0x87B5	BSP	Clear scratch memory (0xFFF) with call to SAL_C	South
0x07B5	BSP	Hang on ERROR.	South
0x87B4	BSP	Search FIT for EFI module with call to PELoader. Get Size, align, and scratch buff size.	South
0x07B4	BSP	Hang on ERROR. Get entry point, and GP value.	South
0x87B3	BSP	Load image by module type (sal_c_module_17).	South
0x07B3	BSP	Hang on ERROR. Get entry point, and GP value.	South
0x87B2	BSP	Flush cache with PAL call.	South
0x07B2	BSP	Hang on ERROR.	South
0x87B1	BSP	Build EFI input parameter table. Get EFI stack, bspstore etc. with EFI call.	South
0x07B2	BSP	Hang on ERROR.	South
0x87B0	BSP	Build EFI input parameter table. Get EFI stack, bspstore etc. with EFI call. Store EFI stack, bspstore etc. with EFI call. Call EFI and that should be end.	South
0x07B0	BSP	Hang on ERROR if OK come back from EFI.	South

7.2.2.4 IA-32 Module

The IA-32 POST codes all have the Most Significant Bit (MSB) cleared by the convention established above in this document. Also, the IA-32 POST codes don't fall into the module definition for Itanium-based platforms above. The codes shown here are consistent with the 7.0 AMI* core.

Table 9. IA-32 POST Codes

Code Value	Module	Display
0x00D0	Power-on delay is starting. Next, the initialization code checksum will be verified.	South
0x00D1	Initializing the Direct Memory Access (DMA) controller, performing the keyboard controller BAT test, starting memory refresh, and entering 4 GB flat mode next.	South
0x00D3	Starting memory sizing next.	South
0x00D4	Returning to real mode. Executing any OEM patches and setting up the stack next.	South
0x00D5	Passing control to the uncompressed code in shadow RAM at E000 0000h.The initialization code is copied to segment 0 and control will be transferred to segment 0.	South
0x00D6	Control is in segment 0. If the system BIOS checksum is bad, will next go to checkpoint code E0h. Otherwise, going to checkpoint code D7h.	South

Code Value	Module	Display
0x00D7	Passing control to the interface module next.	South
0x00D8	The main system BIOS runtime code will be decompressed next.	South
0x00D9	Passing control to the main system BIOS in shadow RAM next.	South
0x0003	Next, checking for a soft reset or a power on condition.	South
0x0005	The BIOS stack has been built. Next, disabling cache memory.	South
0x0006	Uncompressing the POST code next.	South
0x0008	The CMOS checksum calculation is done next.	South
0x000B	Next, performing any required initialization before the keyboard BAT command is issued.	South
0x000C	The keyboard controller input buffer is free. Next, issuing the *BAT* command to the keyboard controller.	South
0x000E	The keyboard controller *BAT* command result has been verified. Next, performing any necessary initialization after the keyboard controller BAT command test.	South
0x000F	The initialization after the keyboard controller BAT command test is done. The keyboard command byte is written next.	South
0x0010	The keyboard controller command byte is written. Next, issuing the Pin 23 and 24 blocking and unblocking commands.	South
0x0011	Check for INS key pressed. Get POST info.	South
0x0012	Disable DMA controllers 1 and 2 and interrupt controllers 1 and 2.	South
0x0013	The video display has been disabled. Port B has been initialized. Next, initializing the chipset.	South
0x0014	The 8254 timer test will begin next.	South
0x0019	The 8254 timer test is over. Starting the memory refresh test next.	South
0x001A	The memory refresh line is toggling. Check the 15-second on/off time next.	South
0x0023	Read the 8042 input-port and disable the MEGAKEY Green PC feature next. Make the BIOS code segment writable and perform any necessary configuration before initializing the interrupt vectors.	South
0x0024	The configuration required before interrupt vector initialization has completed. Interrupt vector initialization is about to begin.	South
0x0025	Interrupt vector initialization is done. Clear the password if the POST DIAG switch is on.	South
0x0027	Any initialization before setting video mode will be done next.	South
0x0028	Initialization before setting the video mode is complete. Configuring the monochrome mode and color mode settings next.	South
0x002A	Bus initialization system, static, and output devices will be done next, if present. Starting LAN redirection, displaying redirection console message.	South
Note that there will be 15-bit post codes in this area. These indicate Device Initialization Manager sub-codes.	The convention for the DIM POST codes is as follows: Port 80 = 0x2A Port 81 = DIM Function number | DI number	South
0x002B	Passing control to the video ROM to perform any required configuration before the video ROM test.	South
0x002C	All necessary processing before passing control to the video ROM is done. Look for the video ROM next and pass control to it.	South
0x002D	The video ROM has returned control to BIOS POST. Perform any required processing	South

Code Value	Module	Display
	after the video ROM had control.	
0x002E	Complete post-video ROM test processing. If the EGA/VGA controller is not found, perform the display memory read/write test next.	South
0x0037	The display mode is set. Display the power on message next.	South
0x0038	Initialize the bus input, IPL, and general devices next, if present.	South
0x0039	Late processor self test. Display bus initialization error messages.	South
0x003A	The new cursor position has been read and saved. Displaying the Hit F2 message.	South
0x0053	The memory size information and the CPU registers are saved. Entering real mode.	South
0x0054	Shutdown was successful. The CPU is in real mode. Disabling the Gate A20 line, and parity next.	South
0x0057	The A20 address line, parity disabled. Adjusting the memory size depending on relocation and shadowing next.	South
0x0058	The memory size was adjusted for relocation and shadowing. Clearing the Hit F2 message.	South
0x0059	The Hit F2 message is cleared. Starting the DMA and interrupt controller test next.	South
0x0060	The DMA page register test passed. Performing the DMA Controller 1 base register test next.	South
0x0062	The DMA controller 1 base register test passed. Performing the DMA controller 2 base register test next.	South
0x0065	The DMA controller 2 base register test passed. Programming DMA controllers 1 and 2 next.	South
0x0066	Completed programming DMA controllers 1 and 2. Initializing the 8259 interrupt controller next.	South
0x007F	TBD.	South
0x0080	Mouse initialization of PS/2 mouse to program the IRQ level to edge triggered or level triggered. The keyboard test has started. Clearing the output buffer and checking for stuck keys. Issuing the keyboard reset command next.	South
0x0082	The keyboard controller interface test completed. Write the command byte and initializing the circular buffer next.	South
0x0083	The command byte was written and global data initialization has completed. Checking for a locked key next.	South
0x0084	Locked key checking is over. Identify ATAPI devices.	South
0x0089	The programming after Setup has completed. Displaying the power on screen message next.	South
0x008B	Init boot devices. Check for and reset mouse.	South
0x008C	Npost adjustments to setup. Form E820 tables. Program SETUP-selected chipset and Sup-IO parameters.	South
0x008D	The Setup options are programmed. Resetting the hard disk controller.	South
0x008E	OEM patches executed. Decompress INT13 module and init ATA & ATAPI devices.	South
0x0093	Done with ATA and ATAPI init. Set printer, RS-232 time out.	South
0x0095	Initializing the bus option ROMs from C800 next. SCSI opt ROM init.	South
0x0091	Configuring the hard disk drive controller. Initializing the CD ROM drive.	South
0x0092	TBD.	South
0x0098	The adaptor ROM had control and has now returned control to BIOS POST. Performing any required processing after the option ROM returned control. Restoring INT10 vector.	South
0x0008	Debugging code.	South

Code Value	Module	Display
0x0099	Configuring the timer data area and printer base address.	South
0x009B	Returned after setting the RS-232 base address. Performing any required initialization before the Coprocessor test next.	South
0x009E	Initialization after the Coprocessor test is complete. Checking the extended keyboard, keyboard ID, and Num Lock key next. Issuing the *Keyboard ID* command.	South
0x00A2	Displaying any soft errors.	South
0x00A3	The soft error display has completed. Setting the keyboard typematic rate.	South
0x00A4	The keyboard typematic rate is set. Programming the memory wait states next.	South
0x00A5	Memory wait state programming is over. Clearing the screen.	South
0x00A7	Performing any initialization required before passing control to the adaptor ROM at E000 next.	South
0x00AE	Setting up DMI structures.	South
0x0020	Talking to BMC	South
0x0022	Talking to BMC.	South
0x00AC	Uncompressing the DMI data and initializing DMI POST.	South
0x00AB	Building the multiprocessor table.	South
0x00AD	Prepare INT10 image. Update the necessary data in different modules.	South
0x00A8	Initialization before passing control to the adaptor ROM at E000h completed. Pass control to the adaptor ROM at E000h.	South
0x00A9	Returned from adaptor ROM at E000h control. Performing any initialization required after the E000 option ROM had control next.	South
0x00AA	Initialization after E000 option ROM control has completed. Displaying the system configuration.	South
0x00B1	Copying any runtime code to specific areas.	South
0x0000	Code copying to specific areas is done. Pass control to EFI.	South

7.2.2.5 EFI Module

The EFI POST codes have been redefined relative to previous platform implementations. Note the addition of the module number and that bit 15 is set, indicating code for Itanium-based platforms.

Table 10. EFI POST Codes

Code Value	Module	Display
0x85F0	Initialize the EFI FW memory map.	South
0x85F1	Set up interrupt vector mappings in interrupt controller for EFI.	South
0x85F2	Memory map installed.	South
0x85F3	Basic EFI services are now functional. Although no devices or variable store support is on line. InitializeLib.	South
0x85F4	Init Watchdog Timer. Initialize support for calling BIOS functions Initialize Bios Int Caller. Install base devices. This would at least include a global_IO device, all NV ram store device(s), and the timer tick. It may optionally include other_IO devices. Install Base Devices.	South
0x85F5	Nv Store installed.	South
0x85F6	Install consoles device, and notify EFI FW to pick and enable a console.	South
0x85F7	Consoles installed.	South

Code Value	Module	Display
0x85F8	Print banner with entry address to make it easy to debug with symbols. Install any devices that are integrated system volume devices.	South
0x85F9	System volumes installed.	South
0x85FA	Init Nv Var Store Mem Set EFIDebug based on NVRAM variable. Set default console environment variables if they are not already set. Install Console Splitter. Print Banner with entry address to make it easy to debug with symbols. Install any other integrated device support. No need to inform FW as devices appear.	South
0x85FB	Say we only support English. Set supported language to English.	South
0x85FC	Create an event to be signaled when ExitBootServices occurs Loop through boot manager and boot maintenance until a boot option is selected. Once the platform code is ready to boot the machine, pass control to the boot manager.	South
(0x0D – 0x0F)	Load shell, tools/debug – MAY NOT APPEAR AFTER BETA BIOS – TBD	
0x5F10	Not used	
0x85E0	Load keyboard driver.	South
0x5FE1	Install VGA class driver.	South
0x5FE2	Initialize VGA Bios Mini Driver.	South
0x5FE3	Get Default Console Mapping.	South
0x5FE4	Init VGA Bios Device Path.	South
0x5FE5	Install Protocol Interfaces for con out, etc.	South
0x5FE6	Debug msg for above.	South
0x5FE7	Update the NVRAM volatile variables with the above devicepath	South
0x5FE8	RESERVED - Platform-specific PC	
0x5FE9	RESERVED - Platform-specific PC	
0x5FEA	RESERVED - Platform-specific PC	
0x5FEB	RESERVED - Platform-specific PC	
0x5FEC	RESERVED - Platform-specific PC	
0x5FED	RESERVED - Platform-specific PC	
0x5FEE	RESERVED - Platform-specific PC	
0x5FEF	RESERVED - Platform-specific PC	

Table 11. ACPI POST Codes

Code Value	Module	Display
0x4F00 – 0x4F	Reserved for ACPI	South

Table 12. SAL Runtime POST Codes

Code Value	Module	Display
0xAFCD	IA-32 Intercept Trap due to an unsupported IA-32 instruction.	South
0xAFE8	Normal SAL Machine Check Handling in Progress.	South
0xAFE9	Could Not Correct MC Error, Halting CPU.	South
0xAFEA	MCA successfully completed, passing control back to PAL (Resume).	South
0xAFEB	Calling OS MCA for Machine Check error handling.	South
0xAFEC	Machine Check Handler Processing Rendezvous Request.	South

0xAFED	OS request for SAL Clear Processor/Platform Error/State Log in progress.	South
0xAFEE	SAL Platform OEM MCA Error Handler In Control.	South
0xAFEF	OS request for SAL Get Processor/Platform Error/State Log in progress.	South
0xAFF0	SAL INIT Handler is in control.	South
0xAFF1	Passing Control to IA-32 OS Init Handler.	South
0xAFF2	Found valid OS_INIT Ep, Passing Control to EM OS Init Handler.	South
0xAFF3	Is a MP platform MCA condition, calling SAL_RENDZ.	South
0xAFF4	Not a MP Platform MCA Init condition.	South
0xAFF5	EM OS with no Init Handler or IA-32OS-BSP detected, Soft Rebooting.	South
0xAFF6	No OS Init Handle Registered, Checking OS Type.	South
0xAFF8	SAL PMI Handler is in Control.	South
0xAFFA	OEM SAL PMI Handler is in Control.	South
0xAFFB	Getting Source of PMI Event.	South
0xAFFC	Power Management PMI Handler is in Control.	South
0xAFFD	Platform Error PMI Handler is in Control.	South
0xAFFE	Platform Flash Management PMI Handler is in Control.	South
0xAFFF	Platform Emulation PMI Handler is in Control.	South
0xAF71	Recover Reliable Update - verifies the bootblock checksum and corrects if possible.	South

7.3 Recovery Port 80 Codes

Table 13. Recovery POST Codes

Code Value	Module	Display
0x8EC0	Recovery Process Started (validating CRC)	South
0x8EC1	Searching for Recovery Media	South
0x8EC2	Loading Recovery File	South
0x8EC3	Validating Recovery File	South
0x8EC4	Unlocking Flash Devices	South
0x8EC5	Erasing Flash Contents	South
0x8EC6	Programming Flash Contents	South
0x8EC7	Validating Flash Contents	South
0x8EC8	Recovery Process Complete	South

8. Beep Codes

During the course of executing POST, there are occassions where fatal problems happen before video is enabled. These fatal errors are conveyed with the use of the speaker via encoded beeps, coupled with post debug codes.

Since the duration of the display-less POST execution is relatively short, there are fewer beep codes than displayed error codes.

In order to extend the useful range of the beep codes, without the need to have dozens of codes, the beeps are classified and the distinction within class is made via the post debug card.

Table 14. Error Beep Codes

Beeps	Error message	Description
3	Memory failure	Memory test failure. See table below for additional error information.
4	System timer	System timer is not operational.
5	Processor failure	Processor failure detected.
7	Processor exception interrupt error	The processor generated an exception interrupt.
8	Display memory read/write error	The system video adapter is either missing or its memory is faulty. This is not a fatal error.
9	ROM checksum error	System BIOS ROM checksum error.
11	Invalid BIOS	General BIOS ROM error.

Table 15. POST Memory Beep Error Codes – Debug Port Encoding List

Beep Code	Debug port error code (lower byte of North I2C debug display)	Meanings
3	CF9Fh	No valid memory was found in the system.
3	CF64h	Mismatched DIMMs in a row, and no valid memory to boot.

8.1.1 Memory Test Failure

8.1.1.1 No Memory Found in the System

This indicates that the memory test has found no valid memory in the system. The system will not boot. An SEL log entry will be made in this case.

8.1.1.2 Mismatched DIMMs Within Single Row Populated

This indicates that only a single row is populated, and that row contains mismatched DIMMs, preventing booting. An SEL log entry will be made in this case.

8.1.2 Recovery Beep Codes

Table 16. Recovery Mode Beep Codes

Beeps	Description
1 short – medium tone	BIOS Flash Update Started
2 short – medium tone	BIOS Flash Update Complete
Repeating – low tone	BIOS Recovery Error Occurred

8.1.3 BMC Beep Code Generation

The BMC generates beep codes upon detection of the failure conditions listed in *Table 17*. Each digit in the code is represented by a sequence of beeps whose count is equal to the digit.

Table 17: BMC Beep Codes

Code	Reason for Beep
1-5-1-1	FRB3 failure (processor failure)
1-5-2-1	Processor: Empty Slot
1-5-2-2	Processor: No Processors
1-5-2-3	Processor: Configuration Error (e.g., VID mismatch)
1-5-4-2	Power fault: DC power unexpectedly lost (power control failures)
1-5-4-3	Chipset control failure
1-5-4-4	Power control fault

Appendix A: Glossary

Term	Definition
ACPI	Advanced Configuration and Power Interface.
ANSI	American National Standards Institute.
ASCII	American Standard Code for Information Interchange. An 8-level code (7 bits plus parity check) widely used in data processing and data communications systems.
ASIC	Application specific integrated circuit.
BERR	Bus Error Signal. This signal can be driven by the platform to interrupt the processor that a platform MCA condition occurred. The processor does not reset any internal state when it sees a BERR condition. The signal causes a global MCA condition. For further information, see the *Itanium™ Processor Family Error Handling Guide.*
BINIT	Bus Initialization Signal. This signal can be driven by the processor or platform to indicate a fatal machine check condition. The processor and platform will reset internal state in order to ensure the firmware code can be fetched and executed. This signal causes a global MCA condition. For further information, see the *Itanium™ Processor Family Error Handling Guide*.
BIOS	Basic Input Output System.
BIST	Built-In Self Test.
BMC	Baseboard Management Controller.
Bridge	Circuitry connecting one computer bus to another, allowing an agent on one to access the other.
BSP	Boot Strap Processor.
byte	8-bit quantity.
CBC	Chassis Bridge Controller. A microcontroller connected to one or more other CBCs. Together they bridge the IPMB buses of multiple chassis.
CHAP	Challenge Handshake Authentication Protocol.
CHS	Cylinder- Head-Sector. An older addressing scheme for accessing physical sectors on hard drives and other storage devices. See **LBA.**
CMCI	Corrected Machine Check Interrupt.
CMOS	In terms of this specification, this describes the PC-AT compatible region of battery-backed 128 bytes of memory, which normally resides on the baseboard.
CPEI	Corrected Platform Event Interrupt.
CVDR	Configuration Values Driven on Reset. A register in the chipset that is accessible by the BMC to control certain system parameters.
DFT	Design for Test. DFT is a set of design rules whose purpose is to improve platform and system testability.
DMA	Direct Memory Access.
DSDT	Differentiated System Description Table. An OEM must supply a DSDT to an ACPI-compatible OS. The DSDT contains the Differentiating Definition Block, which supplies the implementation and configuration information about the base system.
DTLB	Distributed Translation Lookaside Buffer.
DWORD	Double Word, a 32-bit quantity.
EEPROM	Electrically erasable programmable read-only memory.
ECC	Error Correction Code. Refers to a memory system that has extra bit(s) to support limited detection/correction of memory errors.
EMP	Emergency Management Port.
EPS	External Product Specification.
FRB	Fault Resilient Booting.
FRU	Field Replaceable Unit.
GB	1024 MB.

Term	Definition
GPIO	General Purpose I/O.
HSC	Hot-Swap Controller.
Hz	Hertz (1 cycle/second).
I^2C	Inter-integrated circuit bus.
I_2O	Intelligent I/O. An open architecture for the development of device drivers in network system environments
IA	Intel® Architecture.
IBF	Input Buffer.
ICH	I/O Controller Hub.
ICMB	Intelligent Chassis Management Bus.
IERR	Internal Error.
IOP	I_2O compliant-I/O Platforms. These typically contain an I/O processor and I/O subsystem.
IP	Internet Protocol.
IPMB	Intelligent Platform Management Bus. Name for the architecture, protocol, and implementation of a special bus that interconnects the baseboard and chassis electronics and provides a communications media for system platform management information.
IPMI	Intelligent Platform Management Interface. An industry standard that defines standardized, abstracted interfaces to platform management hardware.
ISM	Intel® Server Management.
IR	Infrared.
ITP	In-Target Probe.
KB	Kilobyte=1024 bytes.
KCS	Keyboard Controller Style.
LAN	Local Area Network. A data communications system which allows a number of independent devices to communicate with each other within a moderate size geographic area.
LBA	Logical Block Address. An addressing scheme for accessing sectors on hard drives and other storage devices. The LBA method is preferred over the CHS method (see **CHS**) because it can address more sectors.
LPC	Low Pin Count.
LUN	Logical Unit Number.
MAC	Media Access Control.
MB	Megabyte=1024 Kilobytes.
MD2	Message Digest 2 – Hashing Algorithm.
MD5	Message Digest 5 – Hashing Algorithm – Higher Security.
Ms	Milliseconds.
MSB	Most Significant Bit.
Mux	Multiplexer.
NIC	Network Interface Card.
NMI	Non-maskable Interrupt. The highest priority interrupt in the system, after SMI. This interrupt has traditionally been used to notify the operating system fatal system hardware error conditions, such as parity errors and unrecoverable bus errors.
NVRAM	Non-Volatile RAM.
OBF	Output buffer.
OEM	Original Equipment Manufacturer.
PAL	Processor Abstraction Layer.
PDB	Power Distribution Board.

Term	Definition
PEF	Platform Event Filtering.
PEP	Platform Event Paging.
PERR	Parity Error. A signal on the PCI bus that indicates a parity error on the bus.
PID	Programmable Interrupt Device. The PID is an interrupt controller that provides interrupt steering functions. The PID interfaces include a PCI bus, an APIC bus, and serial IRQ interfaces, and an interrupt input interface.
PIROM	Processor Information ROM. SEEPROM contained in the processor module. Contains information about the processor, such as the core ratio.
PLD	Programmable Logic Device.
PMI	Platform Management Interrupt.
POST	Power-on Self Test.
RAM	Random Access Memory.
RISC	Reduced instruction set computing.
ROM	Read-Only Memory.
RTC	Real-Time Clock. Component of chipset on the baseboard.
SAL	System Abstraction Layer.
SCI	System Control Interrupt. A system interrupt used by hardware to notify the OS of ACPI events.
SDR	Sensor Data Record.
SECC	Single Edge Connector Cartridge.
SEEPROM	Serial Electrically Erasable Programmable Read-Only Memory.
SEL	System Event Log.
SERR	System Error. A signal on the PCI bus that indicates a ‘fatal’ error on the bus.
SMBIOS	System Management BIOS.
SMBus	A two-wire interface based on the I2C protocol. The SMBus is a low-speed bus that provides positive addressing for devices, as well as bus arbitration.
SMI	Server Management Interrupt. SMI is the highest priority non-maskable interrupt.
SMM	Server Management Mode.
SMS	Server Management Software.
SNC	Scalable Node Controller. The north bridge and memory controller (combined) in the 870 chipset.
SNMP	Simple Network Management Protocol.
UART	Universal Asynchronous Receiver/Transmitter.
UDP	User Datagram Protocol.
USB	Universal Serial Bus, a standard serial expansion bus meant for connecting peripherals.
Word	16-bit quantity.

Appendix B: Reference Documents

- *Intelligent Platform Management Interface Specification* v1.5, ©2001, Intel Corporation. http://developer.intel.com/design/servers/ipmi
- *System Management BIOS Reference Specification* v2.3. http://www.dmtf.org/
- *Itanium™ Processor Family Error Handling Guide* (Doc. Number: 249278-002). http://developer.intel.com/
- *Itanium™ System Abstraction Layer Specification* (Doc. Number: 245359-005). http://developer.intel.com/
- *SR870BN4 BIOS External Product Specification* Rev 0.87 (Doc. Number: 11164)
- SR870BN4 Baseboard Management Controller (BMC) External Product Specification Rev 0.87 (Doc. Number: 11365)

Appendix C: Index

A

ACPI, 18, 20, 22, 24, 25, 29
Address, 22
AMI, 25
AP, *See also* Application Processor, 23, 24

B

Baseboard Management Controller, *See also* BMC, 1, 15
BIOS, 1, 3, 9, 13, 14, 15, 18, 22, 24, 25, 26, 27, 28, 29, 31, 32
BIST, 14
BMC, 1, 3, 5, 8, 9, 14, 15, 28, 32
BSP, 19, 20, 21, 22, 23, 24, 25, 30, 31
Built-in Self Test, *See also* BIST, 14
Bus Number, *See also* BUSN, 12

C

Checksum, 13
CMOS, 13, 26, 31
Configuration, 32
Controller, 1, 12, 15, 27

D

DIMM, 9, 10, 18, 19, 21
Direct Platform Control, *See also* DPC, 1
Direct Platform Control, *See* DPC, 1
DMI, 28
DPC, 1
DPC, *See also* Direct Platform Control, 1
Driver, 29

E

ECC, 7, 18
Error, 1, 1, 5, 7, 8, 9, 10, 13, 14, 15, 16, 19, 21, 29, 30, 31, 32
Event Message, 1

F

Fan, 4, 5, 6
Fault Resilient Booting, *See also* FRB, 14
Field Replaceable Unit
See also FRU, 8
FRB, 14
Front Panel, 5
Front Panel reset, 19, 20

I

Initialization, 19, 20, 26, 27, 28
Install, 28, 29
Intel Server Control, *See also* ISC v2.x, *See also* ISC v3.x, 1

L

Legacy, 24
LPC, 10, 12

M

Management Controller, 1, 15
Memory, iv, 3, 5, 6, 9, 10, 11, 13, 14, 16, 19, 20, 21, 23, 25, 28, 31, 32
Message, 1, 13

N

NVRAM, 28, 29

P

Parity Error
See also PERR, 31
Password, 13
PERR, 8, 9, 10, 11
PHP, 5
POST, 1, 5, 13, 18, 19, 20, 21, 24, 25, 26, 27, 28, 29, 30, 31
POST Code, 18, 19, 20, 21, 24, 25, 28, 29, 30
Power Control, 32
Power Unit, 5
Power-on Self-Test
See POST, 1, 5, 13, 18, 19, 20, 21, 24, 25, 26, 27, 28, 29, 30, 31
Processor, 4, 7, 8, 9, 10, 14, 15, 19, 20, 29, 31, 32
Processor Failure, 31

R

Recovery, 20, 21, 30, 32
Reset, 19, 20

S

SBE, 11
SCSI, 4, 5, 12, 27
Security, 4
SEL, 2
SEL, *See also* System Event Log, 1, 2, 3, 8, 9, 10, 13, 15, 32
Sensor, 5
Sensor Event, 1, 2, 5, 7, 8, 9, 10, 11, 15, 30
Sensor, Type, 2, 3, 8
SERR, 8, 9, 10
Server Management, 1
Shadow, 14, 23
Shutdown, 27, 31
SMBIOS, 8, 9, 10
SR460AC4, 1, 3, 9
System Event Log, *See also* SEL, 1, 15
System Management BIOS, 9

T

Temperature, 3, 5

U

USB, *See also* Universal Serial Bus, 12

V

VID, *See also* Vendor Identification, 32
Voltage, 3

W

Watchdog Timer, 28