Intel® Server Platform SR870BH2

Field Error Reference Guide

Revision 1.1

March 2004

Enterprise Platforms and Services Division

Revision History

Date	Revision Number	Modifications
03/2003	0.5	Initial Release.
07/2003	1.0	Production Update
03/2004	1.1	Update

Disclaimers

THIS TEST REPORT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

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 retains the right to make changes to its test specifications at any time, without notice.

The hardware vendor remains solely responsible for the design, sale and functionality of its product, including any liability arising from product infringement or product warranty.

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

Table of Contents

List of Figures

List of Tables

1. Introduction

This document was designed to familiarize the field technician with the SR870BH2 error handling architecture and to provide a quick reference to aid in the diagnosis of system failures.

It presents an overview of applicable EFI based system Management Utilities (SMU), the System Error Log (SEL), Machine Check error handling Architecture (MCA) and error messaging. In additon, many of the error messages have been mapped to a possible point of failure and will include a brief comment regarding debug methodology.

The document is organized by the following chapters:

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, BIOS, and event generators. System sensors can also trigger events that are logged in the SEL.

Some event messages are the result of normal events, such as a normal server boot, or possible minor problems such as a disconnected keyboard. Other events may indicate internal

failures such as a component over-temp condition where thresholds, or ranges of acceptable values have been exceeded. 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 the 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 which is available in the System Management Utility (SMU) that ships on the standard platform resource CD. The system administrator can use this information to monitor the server for warnings and potential critical problems.

3. EFI-Based SELViewer Task

The EFI based SEL Viewer task is only available on the Local version of the SMU. This task will not be available when running the Remote version. The EFI 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:
    - Number of Event
    - Time Stamp
    - Sensor Type and Number
    - Event Description
    - Generator ID

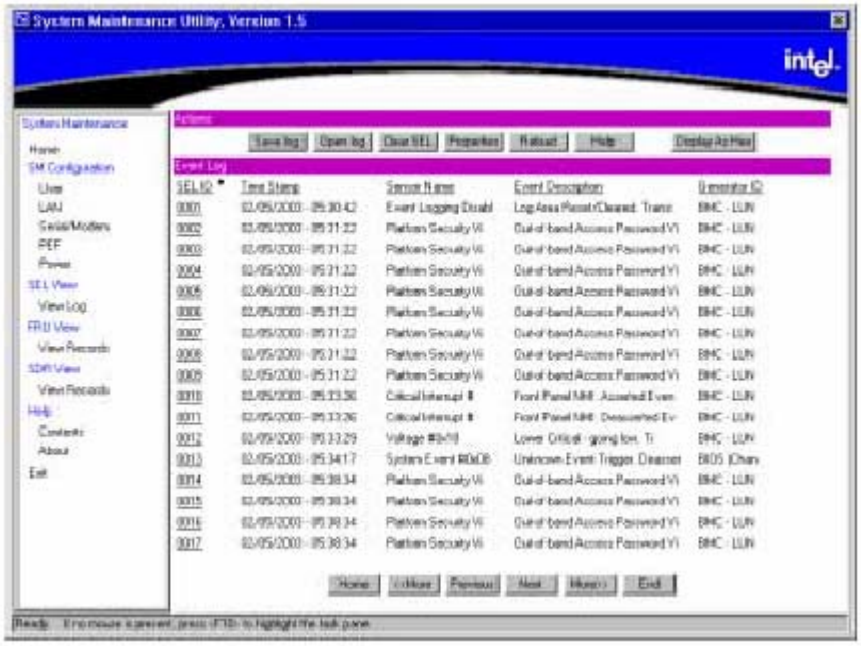

Figure 1. SEL Viewer

4. SR870BH2 SEL Data Tables

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

Table 1. SR870BH2 Generator ID Codes

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

Table 2. SR870BH2 Sensor Codes

Sensor Type	Sensor Number	Sensor Name
01		**Temperature**
	20h	Memory Board Temp
	21h	Memory Board SNC Temp
	22h	PCI Riser SIOH Temp
	23h	Peripheral Board AMB Temp
	24h	PCI Riser board Temp
	25h	CPU Area Temp
	26h	Memory Area Temp
	81h	Processor 1 Temp
	82h	Processor 2 Temp
02		**Voltage**
	10h	MB Bd +1.25V
	11h	MB Bd +1.5V
	12h	MB Bd +1.8V
	13h	MB Bd +3.3V
	14h	MB Bd +3.3V SB
	15h	MB Bd +5V
	16h	MB Bd +12V
	17h	MB Bd –12V
	18h	MB Bd +1.2V
	19h	MB Bd +1.3V
	1Ah	MB Bd –1.5V SB
	1Bh	MB Bd +2.5V
	1Ch	MB Bd +2.5V SB
	1Dh	MB Bd 2 +5V SB
	50h	LVDS SCSI channel 1 terminator 1
	51h	LVDS SCSI channel 1 terminator 2
	52h	LVDS SCSI channel 1 terminator 3
	53h	LVDS SCSI channel 2 terminator 1
	54h	LVDS SCSI channel 2 terminator 2
	55h	LVDS SCSI channel 2 terminator 3

Sensor Type	Sensor Number	Sensor Name
	86h	Proc 1 Power Pod Good
	87h	Proc 2 Power Pod Good
04		**Fan**
	30h	Tach Fan 1
	31h	Tach Fan 2
	32h	Tach Fan 3
	33h	Tach Fan 4
	34h	Tach Fan 5
	35h	Tach Fan 6
	70h	Fan 1 Present
	71h	Fan 2 Present
	72h	Fan 3 Present
	73h	Fan 4 Present
	74h	Fan 5 Present
	75h	Fan 6 Present
05		**Physical Security**
	05h	LAN Leash Lost
06		**Security Violation Attempt**
	04h	Platform Security Violation
07		**Processor**
	80h	Proc 1 Status
	81h	Proc 2 Status
08		**Power Supply**
	60h	Power Supply 1
	61h	Power Supply 2
	62h	Power Supply 3
09		**Power Unit**
	01h	Power Unit Status
	02h	Power Unit Redundancy
0D		**Hot Swap Drive Sensors**
	01h	SCSI BP Temperature
	02h	Hot Swap Drive 1 Status
	03h	Hot Swap Drive 2 Status
	05h	Hot Swap Drive 1 Present
	06h	Hot Swap Drive 2 Present
0F		**POST Error**
	06h	POST Error
10		**Event Logging**
	09h	Event Logging Disabled
12		**System Event**
	12h	OEM System Boot Event PEF Action
13		**Critical Interrupt**
	07h	FP Diag Interrupt (Front Panel SD Init)
15		**Module / Board**
	77h	System Board Interlock

Sensor Type	Sensor Number	Sensor Name
23		**Watchdog**
	03h	BMC Watchdog2
C7		**OEM**
	40h	Fan Boost Mem Board Temp
	41h	Fan Boost Mem Board SNC Temp
	42h	Fan Boost PCI Riser SIOH Temp
	43h	Fan Boost Peripheral Board AMB Temp
	44h	Fan Boost PCI Riser Board Temp
	45h	Fan Boost CPU Area Temp
	46h	Fan Boost Mem Area Temp
	84h	Fan Boost Proc 1 Temp
	85h	Fan Boost Proc 2 Temp

5. SR870BH2 Machine Check Error Handling

This section gives an overview of the implementation of machine check error handling on the SR870BH2 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 www.developer.intel.com.

The goal of MCA is to contain errors and correct as many as possible before they propagate to network or permanent storage. If an error cannot be fixed by the hardware or firmware, and the OS cannot handle it, the machine shall be reset. MCA errors include ECC, BINIT, BERR, SERR, and PERR. These conditions are handled by the BIOS through SAL 3.0-compatible services.

5.1 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 Itanium Machine Check Architecture (MCA) and implemented in the SR870BH2 platform.

5.2 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. 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.

5.3 Error Signaling

There are two classes of error events:

- **Machine Check Error 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.

Machine Check Architecture (MCA): The MCA can be either local or global. In the event of an MCA, the processor will take the exception at instruction boundary with highest priority. In the event of a local abort, the affected processor will enter MCA handling mode. If the event is global, all processors will enter MCA handling mode.

Uncorrectable Error Events:

- **Local MCA:** A local MCA is taken by the processor when it reads data with uncorrectable errors, or receives a hard fail response to a transaction. 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.

- **Global MCA:** A machine check is global when all processors enter machine check. A machine check is global when all processors enter machine check. On the SR870BH2 platform, the method used to get all processors into machine check are the BINIT# and BERR# signals. The processor asserts BINIT#, or there is an assertion of BERR# by the processor or platform. The processor can assert BINIT# on a transaction time-out event. BERR# is asserted by the platform on platform-fatal errors, and can be programmed to assert BERR# when an uncorrectable error is detected on I/O read data.

Correctable Error Events:

- Corrected Machine Check (CMC): 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 Errors (CPE): 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.

5.4 Error Reporting

SR870BH2 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 SR870BH2, based on the MCA records, system events related to Field Replaceable Units (failures) are logged in the BMC SEL. Each MCA record results in the generation of one or more corresponding BMC SEL event(s). 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. All MCAs are logged into NVRAM and the SEL.

The format of the SEL entries is compliant with the IPMI 1.5 specification. The BIOS will log system events and POST error codes. The BIOS will log a boot event to BMC at the end of the POST just before loading EFI. The events logged by the BIOS will follow the IPMI specification.

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

Table 3. 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	

5.5 Thresholding

MCA errors are classified into one of three categories: corrected, recoverable, and fatal. In general, corrected errors will not affect the operation of the system 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 performance degradation 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 these problems, a thresholding algorithm 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 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. Thresholding does not apply to Recoverable or Fatal errors, only corrected errors. On the SR870BH2, 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.

5.6 SEL Event Log Format for Machine Check Errors

The following table shows the machine check errors that will be logged for the SR870BH2, 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 Specification* 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								
PERR	0x31	0x4	0x13		0x6F	0xA4	PCI Bus #	Bit 7:3 -DEV# Bit 2:0 -Func#
SERR	0x31	0x4	0x13		0x6F	0xA5	PCI Bus #	Bit 7:3 -DEV# Bit 2:0 -Func#
PCI Bus PERR/SERR								
PERR	0x31	0x4	0x13	N\A	0x6F	0x84	PCI Bus #	0xFF
SERR	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	0x27
Proc Correctable Logging Disabled	0x31	0x4	0x10	N/A	0x6F	0xF1	0x07	0x20
PCI PERR Logging	0x31	0x4	0x10	N/A	0x6F	0xF1	0x13	0x24
Disabled								
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

6. SR870BH2 PCI Device IDs

The SR870BH2 server has the following PCI devices and slots on the I/O board:

Table 4. 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	
DMH	SNC	0xFF	0x018	1
ICH4	Internal	0	30	0
LPC		0	31	0
IDE Controller		0	31	1
USB Controller1 (1.1)		0	29	0
USB Controller2 (1.1)		0	29	1
Video	Internal	0		
Dual GB NIC	A (embedded)	Dynamic	1	0,1
SCSI Controller	B (embedded)	Dynamic	1	0,1
PCI slot 1	A (133Mhz full-size)	Dynamic	1	
PCI slot 2	B (100Mhz full-size)	Dynamic	1	
PCI slot 3	B (100Mhz full-size)	Dynamic	2	

7. BIOS POST Error Codes and Messages

The following error codes are relevant to the SR870BH2 server. The system BIOS displays POST error messages on the video screen and are also logged in the SEL.

The SR870BH2 BIOS will prompt the user to press a key in case of serious errors.

7.1 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 & Character Attributes	Pause on Boot	Failure
103	CMOS Battery Failure DFLT/RED_BLACK	Yes	Check battery, check / modify CMOS setup and Save (F-10). If failure persists replace Mainboard.
104	CMOS Options not Set DFLT/RED_BLACK	Yes	Check battery, check / modify CMOS setup and Save (F-10). If failure persists replace Mainboard.
105	CMOS Checksum Failure DFLT/RED_BLACK	Yes	Check battery, check / modify CMOS setup and Save (F-10). If failure persists replace Mainboard.
109	Keyboard - Stuck key DFLT/RED_BLACK	Yes	Check for stuck key or replace Keyboard, If failure persists replace Mainboard.
11B	Date/Time not set DFLT/RED_BLACK	Yes	Check battery, check / modify date and time. If failure persists replace Mainboard.
120	NVRAM cleared By jumper DFLT/RED_BLACK	Yes	Check CLR CMOS Jumper J5H3: Normal 1-2, Clear 2-3, If failure persists replace Mainboard.
121	Password clear WARN/YELLOW_BLACK	Yes	Check CLR PASS Jumper J5H2: Normal 1-2, Clear 2-3, If failure persists replace Mainboard.
122	NVRAM cleared By Front panel DFLT/RED_BLACK	Yes	CMOS has been cleared by front panel control sequence. Refer to chapter 11.Clearing CMOS and BIOS recovery. If failure persists suspect / replace front panel 1st or Mainboard 2nd.
140	PCI Error DFLT/RED_BLACK	Yes	PCI bus or device error, remove PCI riser assy Check VHDM connectors for pin damage on PCI riser, remove Add-in adapters, reseat securely and retest. Note (riser must be installed for system to boot- contains core logic) suspect or replace the PCI Riser 1st and the Mainboard 2nd.
141	PCI Memory Allocation Error DFLT/RED_BLACK	Yes	Memory allocation area for PCI device exceeded, remove add-adapter(s) retest. Suspect or replace any add-in adapter(s) 1st, the PCI Riser 2nd and the Mainboard 3rd.

Error Code	Error Message & Character Attributes	Pause on Boot	Failure
142	PCI IO Allocation Error DFLT/RED_BLACK	Yes	PCI I/O Resource allocation for PCI devices has been exceeded - remove add-adapter(s) retest. Suspect or replace any add-in adapter(s) 1st, the PCI Riser 2nd and the Mainboard 3rd.
143	PCI IRQ Allocation Error DFLT/RED_BLACK	Yes	PCI IRQ Resource allocation for PCI devices has been exceeded - remove add-adapter(s) retest. Suspect or replace any add-in adapter(s) 1st, the PCI Riser 2nd and the Mainboard 3rd.
144	Shadow of PCI ROM Failed DFLT/RED_BLACK	Yes	PCI ROM memory allocation area for PCI devices has been exceeded - remove add-adapter(s) retest. Suspect or replace any add-in adapter(s) 1st, the PCI Riser 2nd and the Mainboard 3rd.
145	PCI ROM not found DFLT/RED_BLACK	Yes	PCI ROM memory allocation area for PCI devices has been exceeded or PCI ROM did not have space to load - remove add-adapter(s) retest. Suspect or replace any add-in adapter(s) 1st, the PCI Riser 2nd and the Mainboard 3rd.
146	Insufficient Memory to Shadow PCI ROM DFLT/RED_BLACK	Yes	PCI Option ROM area exceeded - remove add-adapter(s) retest. Suspect or replace any add-in adapter(s) 1st, the PCI Riser 2nd and the Mainboard 3rd.
8100	Processor 01 failed BIST WARN/YELLOW_BLACK	Yes	Processor failed to initialize in time, Refer to “Fault Resilient Boot” in Debug Methodology and failure Isolation section.
8101	Processor 02 failed BIST WARN/YELLOW_BLACK	Yes	Processor failed to initialize in time, Refer to “Fault Resilient Boot” in Debug Methodology and failure Isolation section.
8110	Processor 01 Internal error (IERR) WARN/YELLOW_BLACK	Yes	Refer to “Processor” in Debug Methodology and failure Isolation section.
8111	Processor 02 Internal error (IERR) WARN/YELLOW_BLACK	Yes	Refer to “Processor” in Debug Methodology and failure Isolation section.
8120	Processor 01: Thermal trip failure. WARN/YELLOW_BLACK	Yes	Processor 01 has exceeded the thermal diode temperature limit resulting in a thermal trip event, check for airflow obstructions to fans & heat sinks. Ensure processor air duct is present and secured. The system is designed to run heavy loads up to 35C or 96F ambient. However, Under normal (i.e. no fan failure) conditions, the CPU should not throttle until 42C/ 107F. (With a fan failure, throttling could possibly occur at about 27C) Thermal trip should occur at 50C /120F, thermal trip will shut down the processor or system. If this occurs and the ambient temp is far cooler - Refer to “Processor” in Debug Methodology and failure Isolation section.
8121	Processor 02: Thermal trip failure. WARN/YELLOW_BLACK	Yes	Processor 02 has exceeded the thermal diode temperature limit resulting in a thermal trip event, check for airflow obstructions to fans & heat sinks. Ensure processor air duct is present and secured. The system is designed to run heavy loads up to 35C or 96F ambient. However, Under normal (i.e. no fan failure) conditions, the CPU should not throttle until 42C/ 107F. (With a fan failure, throttling could possibly occur at about 27C) Thermal trip should occur at 50C /120F, thermal trip will shut down the processor or system. If this occurs and the ambient temp is far cooler - Refer

Error Code	Error Message & Character Attributes	Pause on Boot	Failure
			to “Processor” in Debug Methodology and failure Isolation section.
8130	Processor 01: Disabled WARN/YELLOW_BLACK	Yes	Processor failed to initialize in time, Refer to Refer to “Fault Resilient Boot” in Debug Methodology and failure Isolation section.
8131	Processor 02: Disabled WARN/YELLOW_BLACK	Yes	Processor failed to initialize in time, Refer to Refer to “Fault Resilient Boot” in Debug Methodology and failure Isolation section.
8140	Processor 01: failed FRB level 3 timer WARN/YELLOW_BLACK	Yes	Processor failed: Fault Resilient Boot Timer (FRB) expired. Refer to Refer to “Fault Resilient Boot” in Debug Methodology and failure Isolation section.
8141	Processor 02: failed FRB level 3 timer WARN/YELLOW_BLACK	Yes	Processor failed: Fault Resilient Boot Timer (FRB) expired. Refer to Refer to “Fault Resilient Boot” in Debug Methodology and failure Isolation section.
8150	Processor 01: failed initialization on last boot WARN/YELLOW_BLACK	Yes	Processor failed to initialize in time, Refer to Refer to “Fault Resilient Boot” in Debug Methodology and failure Isolation section.
8151	Processor 02: failed initialization on last boot WARN/YELLOW_BLACK	Yes	Processor failed to initialize in time, Refer to Refer to “Fault Resilient Boot” in Debug Methodology and failure Isolation section.
8192	L3 cache size mismatch WARN/YELLOW_BLACK	No	BIOS compared Processors & determined an L3 cache mismatch; both processors should run with different cache sizes. If both processors are identical - Refer to “Processor” in Debug Methodology and failure Isolation section.
8193	CPUID, Processor Steppings are different WARN/YELLOW_BLACK	No	BIOS compared Processors & determined a mismatch, both processors may run – If different speeds they will be Performance restricted, both will run at lower of the two speeds, If incompatible - one processor may be disabled. Check stepping & P/N information – Dissimilar stepping processors are not supported. Refer to “Processor” in Debug Methodology and failure Isolation section.
8196	Processor Models are Different DFLT/RED_BLACK	Yes	BIOS compared Processors & determined a mismatch, both processors may run – If different speeds they will be Performance restricted, both will run at lower of the two speeds, If incompatible - one processor may be disabled. Check stepping & P/N information – Dissimilar stepping & mixed family processors are not supported. Refer to “Processor” in Debug Methodology and failure Isolation section.
8197	Processor speeds mismatched DFLT/RED_BLACK	No	BIOS compared Processors & determined mismatched speeds, both processors will be Performance restricted, both will default to run at lower of the two speeds. If both processors are identical - Refer to “Processor” in Debug Methodology and failure Isolation section.
8210	Processor 1 Late Self Test Failed: Performance restricted DFLT/RED_BLACK	Yes	Failure identified in late self-test, Processor 1 will be Performance restricted. Refer to “Processor Late Self test” in Debug Methodology and failure Isolation section.
8211	Processor 2 Late Self Test Failed: Performance restricted DFLT/RED_BLACK	Yes	Failure identified in late self-test, Processor 2 will be Performance restricted. Refer to “Processor Late Self test” in Debug Methodology and failure Isolation section.
8220	Processor 1 Late Self Test Failed: Functionally restricted	Yes	Failure identified in late self-test, Processor 1 will be Functionally restricted. Refer to “Processor

Error Code	Error Message & Character Attributes	Pause on Boot	Failure
	DFLT/RED_BLACK		Late Self test" in Debug Methodology and failure Isolation section.
8221	Processor 2 Late Self Test Failed: Functionally restricted DFLT/RED_BLACK	Yes	Failure identified in late self-test, Processor 2 will be Functionally restricted. Refer to "Processor Late Self test" in Debug Methodology and failure Isolation section.
8230	Processor 1 Late Self Test Failed: Catastrophic failure DFLT/RED_BLACK	Yes	Failure identified in late self-test, Processor 1 will be disabled. Refer to "Processor Late Self test" in Debug Methodology and failure Isolation section.
8231	Processor 2 Late Self Test Failed: Catastrophic failure DFLT/RED_BLACK	Yes	Failure identified in late self-test, Processor 2 will be disabled. Refer to "Processor Late Self test" in Debug Methodology and failure Isolation section.
8300	Baseboard Management Controller failed to function DFLT/RED_BLACK	Yes	Check to see that the jumper at J5H4 is in position 1-2 =normal, (position 2-3 is update mode). If Issue persists replace Mainboard.
8306	OS boot watchdog timer failure DFLT/RED_BLACK	Yes	System exceeded 6-minute watchdog timer on boot cycle. Refer to "Watch dog timer" for detail.
84F3	Baseboard Management Controller in Update Mode DFLT/RED_BLACK	Yes	Check to see that the jumper at J5H4 is in position 1-2 =normal, (position 2-3 is update mode). If Issue persists replace Mainboard.
84FF	System Event Log Full DFLT/RED_BLACK	Yes	The System Event Log is Full, Save or Clear SEL, See section on "Thresholding"
8500	Multi-bit Error Detected Row1. Row mapped out WARN/YELLOW_BLACK	Yes	The multi-bit error is detected on a DIMM(s) in Row 1; Row disabled, Refer to "Memory" in Debug Methodology and failure Isolation section.
8501	Multi-bit Error Detected Row2. Row mapped out WARN/YELLOW_BLACK	Yes	The multi-bit error is detected on a DIMM(s) in Row 2; Row disabled, Refer to "Memory" in Debug Methodology and failure Isolation section.
8504	Persistent Single-bit Error Detected Row1. Row mapped out. WARN/YELLOW_BLACK	Yes	Issue with DIMM address or data line likely affecting multiple DIMMs (otherwise ECC should recover) in Row 1; Row disabled, Refer to "Memory" in Debug Methodology and failure Isolation section.
8505	Persistent Single-bit Error Detected Row2. Row mapped out. WARN/YELLOW_BLACK	Yes	Issue with DIMM address or data line likely affecting multiple DIMMs (otherwise ECC should recover) in Row 2; Row disabled, Refer to "Memory" in Debug Methodology and failure Isolation section.
8508	Memory Mismatch detected Row1. Row mapped out. WARN/YELLOW_BLACK	Yes	Issue with DIMM SPD value in Row 1; Row disabled, Refer to "Memory" in Debug Methodology and failure Isolation section.
8509	Memory Mismatch detected Row2. Row mapped out. WARN/YELLOW_BLACK	Yes	Issue with DIMM SPD value in Row 2; Row disabled, Refer to "Memory" in Debug Methodology and failure Isolation section.
850C	DIMM1 defective. WARN/YELLOW_BLACK	Yes	Issue with DIMM 1=J9J3, Refer to "Memory" in Debug Methodology and failure Isolation section.
850D	DIMM2 defective. WARN/YELLOW_BLACK	Yes	Issue with DIMM 2=J9J1, Refer to "Memory" in Debug Methodology and failure Isolation section.
850E	DIMM3 defective. WARN/YELLOW_BLACK	Yes	Issue with DIMM 3=J9D3, See memory in Debug Methodology and failure Isolation section.
850F	DIMM4 defective. WARN/YELLOW_BLACK	Yes	Issue with DIMM 4=J9D1, See memory in Debug Methodology and failure Isolation section.
8510	DIMM5 defective. WARN/YELLOW_BLACK	Yes	Issue with DIMM 5=J9J2, See memory in Debug Methodology and failure Isolation section.
8511	DIMM6 defective. WARN/YELLOW_BLACK	Yes	Issue with DIMM 6=J8J1, See memory in Debug Methodology and failure Isolation section.
8512	DIMM7 defective. WARN/YELLOW_BLACK	Yes	Issue with DIMM 7=J9J2, See memory in Debug Methodology and failure Isolation section.
8513	DIMM8 defective.	Yes	Issue with DIMM 8=J8D1, See memory in Debug

Error Code	Error Message & Character Attributes	Pause on Boot	Failure
	WARN/YELLOW_BLACK		Methodology and failure Isolation section.

8. Debug Methodology & Failure Isolation

8.1 Memory

If the memory test finds any bad DIMM(s) (defined as mismatched DIMMs within a row, multi-bit errors [MBE] detected within a DIMM, single-bit [SBE] non-transient errors within a DIMM), the entire associated row will be mapped out and autoscan will not include any memory that is mapped out. The memory test can isolate persistent or non-transient single-bit and multi-bit errors to the defective DIMM and will make that information available.

If mismatched or bad DIMMs are found during the initial geometry check, the bad row will be logged to the System Event Log (SEL). If both rows are determined to be bad, the system will not boot, and the bad row(s) will be logged to SEL. Additionally, the system will emit beep codes as documented in the post and error codes section.

Assuming that there is at least one good row, the bad row will be reported as an error when video is available will be logged to the SEL, and the system will continue to boot.

8.1.1 Memory Debug Methodology

Remove memory DIMM, look for bent pins or obvious sign of contamination on DIMM or inside DIMM site - Reseat the memory; swap ROW one with ROW two (2nd set of 4 DIMMS) reconfigure to minimum memory configuration (4 DIMMs in one ROW, DIMM sites 1,2,3,4 must be populated at a minimum).

NOTE: Even though DIMMs are numbered and populated consecutively, DIMM Sites are not physically consecutive. Refer to the silkscreen on the Mainboard near each DIMM site – first 4 DIMMS in order below.

DIMM Sites 1=J9J3, 2=J9J1, 3=J9D3, 4=J9D1 FIRST. Sites 5=J9J2, 6=J8J1, 7=J9J2, 8=J8D1 are optional.

Exchange the memory, one ROW at a time, with a complete set of known good memory. If failure symptoms persist, suspect the motherboard. If it passes, replace the suspect memory, one at a time, into their original sockets, until the Failure recurs (if replaced one by one - DDR must be same size, die & vendor).

8.1.2 Memory COMPONENT Isolation

DDR 266 Memory can be replaced, one by one, with same mfg P/N (same size, die & vendor) A complete row must be replaced in sets of 4 (minimum config is 4 x 256= 1GB). Populate DIMM Sites 1=J9J3, 2=J9J1, 3=J9D3, 4=J9D1 FIRST. Sites 5=J9J2, 6=J8J1, 7=J9J2, 8=J8D1 are optional. If failure persists replace Mainboard.

8.2 Processor

8.2.1 Processor Debug Methodology:

1) Enter Setup, startup options, select processor retest – Save & Exit (F10) reset system.

2) Run Platform Diagnostic test (located on Resource CD); if error persists perform the following checks and steps:

- Turn off and remove AC source power; remove top cover and processor air duct.

- Reseat DC harness to power pod. Check to see that processor is locked (locking flag or tab on the side of processor socket is visible).

- Check to see that the processor screws (4) are tightened to 6-inch pounds (Use T-15 Torx – do not over tighten). If tab is not visible, loosen the (4) captive mounting screws and lock the processor (if tab is visible = locked, use 2.5mm Allen clockwise ¼ turn to lock) tighten the (4) mounting screws replace covers, reconnect AC source, re-boot / re-run test.

If failure symptom persists:

- Turn off and remove AC source power; remove top cover and processor air duct.
- Remove DC power harness from power pod, remove power pod (loosen 4 captive screws / slide pod away from processor to disengage).
- Remove and inspect pins on processor (loosen 4 captive screws on processor, turn lock tab counterclockwise ¼ turn, “tab not visible”, lift processor to disengage).
- Reseat processor & power pod (repeat above steps in reverse order) and re-run tests.
- Replace / swap power pods or BSP (1st) with APP (2nd) processor.
- Replace covers, reconnect AC source, restart system.
- Re-run test in single processor mode (socket 1 must be populated to boot, Socket 2 will auto terminate).

8.2.2 Processor COMPONENT Isolation

If after reseating or swapping component positions, the suspect component cannot be located (i.e. DC harness, Power pod or Processor) and the failure persists - Replace the Mainboard.

8.3 Processor - Late Self-test

Processor late self-test helps BIOS to determine whether the processors present in the system are healthy enough to boot and run the OS. Once the system memory is initialized, BIOS SAL calls PAL to perform “late self test” on the processor(s) present in the system. The possible late self-test results for each processor are:

- **Functionally restricted** – machine will continue through POST until the sign-on banner and memory test results are displayed. At this time the late self-test will display a message as documented in the next section.

- **Performance restricted** – machine will continue through POST until the sign-on banner and memory test results are displayed. At this time the late self-test will display a message as documented in the next section.

- **Catastrophic failure** – Itanium® 2 processor does not return from PAL in this failure case. The BIOS assumes this as the last condition after eliminating other possibilities (in the order Healthy, Functionally, Performance Restricted). – Refer to the "Processor" section of Debug Methodology and COMPONENT Isolation.

- **Healthy** - If the late self test result is healthy, the system continues the boot as expected and no messages are displayed.

It is at the point of displaying the late self-test errors that any errors encountered are logged to SEL. This is done because the BIOS will have to reset the system early in POST, well before the POST error manager is called, in order to selectively take a failed processor off-line.

8.3.1 Late Self-test Display

Immediately after the BIOS sign-on banner information is displayed, if any processor late self-test error is encountered, the BIOS displays the following message:

Errors found in the processor late self-test. Please wait while the failed processor is disabled for the next boot. System will reset automatically. After this message is displayed, the system will be reset.

8.3.2 Late Self-test Usage Notes

Because the late self-test relies on encapsulated PAL code, there are certain conditions under which the test will operate. These are listed below.

- Only one processor will be disabled per boot cycle.

- On the next boot, the unhealthy processor is not included in the system boot.

- If errors occur during any processor late self-test, the POST error manager will not be displayed on the boot cycle in which the error is detected.

- The POST error manager will only display the fact that a processor is disabled and this will occur on the boot cycle after the processor is disabled. In order to determine if the processor is disabled because of late self-test errors, the SEL will have to be referenced.

8.4 Watch Dog Timer

The BIOS Setup offers a control item that allows the OS load watchdog timer to be enabled or disabled. The default for the OS load watchdog timer function is **disabled**. The server BIOS will support the OS load watchdog timer. This may also be referred to as FRB-4, although the term "OS load watchdog timer" is more accurate, as this timer has no FRB-related connection to disabling processors.

8.4.1 Watch dog timer Debug Methodology:

The Watchdog Timer can be enabled / modified in system setup (Enter Setup / startup options) this menu item is also located under the System Management Submenu – Platform Event Filters (PEF). The watchdog timer provides a 'timer use' field that indicates the current use assigned to the watchdog timer. If enabled and depending on selections configured the timer allows the user to:

- Log an event to SEL upon expiration of the OS load watchdog timer.
- Select the timeout action to be hard reset, and pre-timeout interrupt type to none.
- Set the pre-timeout interval to zero; the pre-timeout action occurs concurrently with the timeout action.
- Program the countdown value to selectable seconds.

8.4.2 Watch dog timer failure Isolation

In most cases this issue is caused by the boot device time out, bad / missing boot block or a corrupted file or file system which causes the system to loop or wait and the timer to trigger (approx. 6 minutes). Depending on boot device / method, the Hard disk subsystem and related component should be reviewed and tested.

8.5 Fault Resilient Boot (FRB)

The BIOS and BMC firmware provide a feature to guarantee that the system boots, even if one or more processors fails during POST. The BMC contains two FRB timers that can be configured to reset the system upon time out.

8.5.1 FRB3 – BSP Reset Failures.

The first timer (FRB-3) starts counting down when the system comes out of hard reset. If the Bootstrap Processor (BSP) successfully resets and begins executing, the BIOS disables the FRB-3 timer in the BMC and the system continues executing POST. If the timer expires because of the BSP failure to fetch or execute BIOS code, the BMC resets the system and disables the failed processor. In this failing scenario, the BMC continues to change the BSP until the BIOS successfully disables the FRB-3 timer. The BMC sounds beep codes on the system speaker if it fails to find a good processor. It will continue to cycle until it finds a good processor. The process of cycling through all the processors is repeated upon system reset or power cycle. The duration of the FRB-3 timer is 6 minutes.

8.5.2 FRB2 – BSP POST Failures.

The second timer (FRB-2) is set for approximately 6 minutes (pending tuning) by BIOS and is designed to guarantee that the system completes POST. The FRB-2 timer is enabled just before the FRB-3 timer is disabled to prevent any "unprotected" window of time. Before the option ROMs are initialized, or if the password prompt is displayed, the BIOS disables the FRB-2 timer. Finally, if the system is set to perform a processor late self-test, the FRB-2 timer will be suspended.

If the system hangs during POST, before the BIOS disables the FRB-2 timer, the BMC generates an asynchronous system reset (ASR). The BMC retains status bits that can be read by BIOS later in the POST for the purpose of disabling the previously failing processor, logging the appropriate event into the SEL, and displaying an appropriate error message to the user.

8.5.3 FRB1 – BSP Self-Test Failures.

In addition to FRB-3 and FRB-2 timers, the BIOS provides FRB-1. Early in POST, the BIOS checks the Built-in Self Test (BIST) results of the BSP. If the BSP fails BIST, the BIOS requests the BMC to disable the BSP. The BMC disables the BSP, selects a new BSP and generates a system reset. If there is no alternate processor available, the BMC beeps the system speaker and enters into “final desperation mode”, a scheme whereby the system will attempt to boot in spite of failed processor(s).

The BIOS and BMC implement additional safeguards to detect and disable the application processors (AP) in a multiprocessor system. If an AP fails to complete initialization within a certain time, it is assumed to be nonfunctional. If the BIOS detects that an AP has failed BIST or is nonfunctional, it requests the BMC to disable that processor. When the BMC disables the processor and generates a system reset, the BIOS will not see the bad processor in the next boot cycle. The failing AP is not listed in ACPI APIC tables, and is invisible to the OS.

8.5.4 FRB Debug Methodology:

All the failures (FRB-3, FRB-2, FRB-1, and AP failures) including the failing processor are recorded into the SEL. The FRB-3 failure is recorded automatically by the BMC, while the FRB-2, FRB-1, and AP failures are logged to the SEL by the BIOS. In the case of an FRB-2 failure, some systems will log additional information into the OEM data byte fields of the SEL entry. This additional data indicates the last POST task that was executed before the FRB-2 timer expired. This information may be useful for failure analysis.

The BIOS and BMC maintain failure history for each processor in nonvolatile storage. This history is used to store a processor’s track record. Once a processor is marked “failed,” it remains “failed” until the user forces the system to retest the processor by entering BIOS Setup and selecting the “Retest processors” option. The BIOS reminds the user about a previous processor failure during each boot cycle until all processors have been retested and successfully passed the FRB tests or AP initialization.

It is possible for all the processors in the system to be marked bad. If all the processors are bad, the system, in final desperation mode, does not alter the BSP and attempts to boot from the original BSP. Again, error messages are displayed on the console and errors are logged in the SEL against a failing or non-healthy processor, with the exception of the single processor case, where the error will be logged, but failing desperation mode, there will be no video display.

If the user replaces a processor that has been marked bad by the system, the system must be informed about this change by running BIOS Setup and selecting the processor retest option.

User selection of the Retest Processor option, in BIOS Setup, results in the BIOS and BMC clearing the Processor failure history from their respective non-volatile storage.

There are three possible states for each processor slot:

- Processor installed (status only; indicates processor has passed BIOS POST).
- Processor failed. The processor may have failed FRB-2, FRB-3, or BIST, and it has been disabled.
- Processor not installed (status only, indicates the processor slot has no processor in it).

8.5.5 FRB Failure Isolation

The issue may be caused by add-in adapter resource contention causing the timer to expire and trigger an FRB, the system should be brought to a minimum config to eliminate variables and restarted w/processor re-test (Enter BIOS / startup options). If the failure persists Refer to “Processor” in Debug Methodology and failure Isolation section.

9. 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 are employed by the SR870BH2 BIOS.

- The first and most common method is Audible, encoded beep sequences emitted by the PC speaker when an error is encountered. Beep codes are employed before the display screen is enabled, and generally indicate *fatal* errors. Beep codes are coupled with special port 80 error codes.

- The second method is to display an error message to the display screen after the Video has been initialized.

- The third method is to display port 80/81 codes to an I_2C* adapter connected to the Mainboard. This method is described as an option but is beyond the scope of this document.

9.1 North and South Port 80/81 Cards

The SR870BH2 server is designed to use an I_2C port 80 card to display POST status codes. The Port 80 card is 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. Since the I2c port POST decoders are considered specialized test equipment, error codes that require the use on of a Port 80 card are beyond the scope of this document.

10. Beep Codes

During the course of executing POST, there are occasions where fatal problems may occur 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. Bolded and italicized items are not used on SR870BH2 BIOS.

Table 5. Error Beep Codes

Beep count	Error message	Description	Course of action / Possible Failure
3	Memory failure - Memory test failure.	1) No valid memory was found in the system. 2) Mismatched DIMMs in a row, Row disabled and no valid memory to boot.	See “Memory” in Debug Methodology and failure Isolation section.
4	System timer	System timer is not operational	See “Processor” in Debug Methodology and failure Isolation section. If symptom persists replace Mainboard.
5	Processor failure	Processor failure detected	See “Processor” in Debug Methodology and failure Isolation section.
7	Processor exception interrupt error	The CPU generated an exception interrupt.	See “Processor” in Debug Methodology and failure Isolation section.
8	Display memory read/write error	The system video adapter is either missing or its Memory is faulty.	Check Add-in video adapter if used, if onboard is used Video DRAM may be failing, replace Mainboard.
9	ROM checksum error	System BIOS ROM checksum error	Corrupted BIOS, Clear CMOS or perform BIOS recovery –refer to “Clearing CMOS and BIOS Recovery” section
11	Invalid BIOS	General BIOS ROM error	Corrupted BIOS, Clear CMOS or perform BIOS recovery –refer to “Clearing CMOS and BIOS Recovery” section

10.1 Recovery Beep Codes

These audible codes describe the progress of a BIOS recovery attempt from a recovery CD refer to chapter 11 *Clearing CMOS and BIOS recovery* for detail and steps to perform this process.

Table 6. 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, or no recovery media found

10.2 BMC Beep Code Generation

The BMC generates beep codes upon detection of the failure conditions, each digit in the code is represented by a sequence of beeps whose count is equal to the digit.

Table 7. BMC Beep Codes

Code	Reason for Beep	Course of action
1-5-1-1	FRB3 failure (processor failure)	See “FRB” in Debug Methodology and failure Isolation section.
1-5-2-1	Processor: Empty Slot	No processor found in socket 1, See “Processor” in Debug Methodology and failure Isolation section.
1-5-2-2	Processor: No Processors	No processor found in socket 1, See “Processor” in Debug Methodology and failure Isolation section.
1-5-2-3	Processor: Configuration Error (e.g., VID mismatch)	General Processor Issue, See “Processor” in Debug Methodology and failure Isolation section.
1-5-4-2	Power fault: DC power unexpectedly lost (power control failures)	DC Voltage error, (1) check TPS power supply (s) for amber lights – replace if found (2) Check / replace power cage.
1-5-4-3	Chipset control failure	
1-5-4-4	Power control fault	DC Voltage error, or dead short (MOST Common cause) i.e. DC to GND (1) check TPS power supply (s) for amber lights – replace if found (2) Check Flex Cable for damage (nick in housing) or poor connection – reseat cable or replace, check all board to board connections i.e. VHDM connectors (Riser to Mainboard) carefully look for Bent

Code	Reason for Beep	Course of action
		or flattened pins at end of connector near fans (3) Check / replace power cage.

11. Clearing CMOS and BIOS recovery

11.1 CMOS Clear

The CMOS must be cleared after the BIOS is updated. If using the automated System Update Package (SUP), the script will automatically clear the CMOS after the BIOS is updated. However if SUP is not used, the CMOS must be cleared manually. Clearing the CMOS involves

- Restarting the server with the new jumper setting
- Restoring the jumper setting to its original position
- Restarting the server a final time

As an alternative, the CMOS clear button sequence can be used from the front panel. See the instructions that follow.

To clear the CMOS using the front panel button sequence follow these steps:
Power down the server by pressing and holding down the power button on the front control panel. Hold down the power button down for several seconds.

- Assure that the system is off, but AC power is connected (5 V standby available).
- Assure that the CMOS clear jumper is in the 'not clear' position.
- Hold down the Reset button for at least 4 seconds. Without letting up on the Reset button, press the On/Off button.
- Release both the On/Off button and Reset button simultaneously. The system will emit one beep.
- Power on the server by pressing and holding the power button on the front control panel.

To clear the CMOS using the jumper on the main board follow these steps:

- Power down the server by pressing and holding down the power button on the front control panel. Hold down the power button for several seconds.
- Unplug both power cords from the server.
- Remove the top cover from the chassis.
- Move the jumper at J5H3 from pins 1-2 to pins 2-3
- Plug in the power cords.
- Power on the server by pressing the power button on the front control panel.

- Wait for the message NVRAM cleared by jumper to appear. When the following message appears, press <F1> to load the defaults:
- Press <F1> to load defaults or <F2> to run SETUP or <ESC> to continue
- Power down the server by pressing and holding the power button on the front control panel. To do so, hold down the power button for several seconds.
- Unplug both power cords from the server.
- Move the jumper at J5H3 from pins 2-3 to pins 1-2.
- Install the chassis cover.
- Plug the power cords back in.
- Power on the server by pressing and holding the power button on the front control panel.

11.2 BIOS Recovery Mode

The BIOS Recovery Mode permits re-flashing the BIOS when the flash ROM has been corrupted. The sequence of events for automatic recovery is:

- Insert recovery media and reset the system.
- One beep indicates recovery media valid, and flash update started.
- Approximately two minutes later, two beeps indicate flash update complete.
- System automatically resets and starts the new BIOS.

NOTE: BIOS recovery requires an “El Torito” formatted CD; alternate forms of removable media including USB devices are not supported and will result in a continuous beep code (approximately 1 beep every 2 seconds until the system is powered down).

- The BIOS Recovery Mode is initiated using the following procedure:
- Unzip the recovery image and copy the SR870BH2.REC file onto the CD. SR870BH2.REC should be the only file on the disk.
- Power down the server.
- Unplug both power cords from the server.
- Remove the top cover from the chassis.
- Move the jumper at J5H1 Labeled ‘RCV BOOT’ from pins 1-2 to pins 2-3.

- Reconnect the AC power and switch server power on. CD Recovery activity begins. One full beep is emitted as the server begins to load SR870BH2.REC from disk to memory.
- Wait two minutes. Two beeps indicate the BIOS recovery has completed successfully.
- Remove the CD and power down the server.
- Unplug both power cords from the server.
- Move the jumper at J5H1 from pins 2-3 to pins 1-2.
- Install the chassis cover.
- Reconnect AC power and power on the server.

Follow any other instructions in the BIOS release notes.

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	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 Look-aside 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.
GPIO	General Purpose I/O.
HSC	Hot-Swap Controller.
Hz	Hertz (1 cycle/second).
I2C	Inter-integrated circuit bus.
I2O	Intelligent I/O. An open architecture for the development of device drivers in network system environments
IA	Intel® Architecture.
IBF	Input Buffer.

Term	Definition
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.
MCA	Machine Check Architecture – see SR870BH2 BIOS EPS for more detail
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.
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.

Term	Definition
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/