Ampere® Altra® Family 64-Bit Multi-Core Processor

SoC Baseboard Management Controller Interface Specification

February 9, 2023

Document Issue 1.42

AMPERE.

Contents

Figures

Tables

1. Overview

This specification describes the high-level interface for supporting a Baseboard Management Controller (BMC) and focuses on high-level hardware and software requirements to support a BMC on a platform containing Ampere Computing® Altra® or Altra® Max processors, also call Systems-on-Chip (SoCs).

Note: In this specification, the term “processor” refers to both Altra and Altra Max processors.

BMC designs generally result from system-level architectural requirements. In this document, statements are sometimes made about what an implementation “should” do. Such statements are intended as guidelines, consistent with common practice, and do not represent requirements of the processor.

Note: This specification is subject to change.

2. Hardware Interfaces

2.1 Processor to BMC Hardware Connectivity

The generic hardware connectivity requirements between one or two processors, including processors in dual-socket (2P) systems, and the BMC include:

- An SMBus-based I2C_4 bus must be configured as a master and attached to an Inter-Integrated Circuit (I^2C) slave port of the BMC. This link between the processor and the BMC is for host-initiated requests. The I2C_4 bus is the SMBus System Interface (SSIF) between the processor and the BMC. The processor I2C_4 ALERT_L[1] input should connect to the BMC output (I2C4_ALERT_L), which functions as the BMC_ALERT_L signal from the BMC. The BMC uses this signal to alert the processor when the BMC sends a message to the host over SSIF. In 2P systems, only the master socket I2C_4 bus and I2C_4 ALERT_L signals connect to the BMC.
- A SMBus-based I2C_3 bus must connect the processor to another I^2C bus of the BMC. The processor I2C_3 port is configured as an I^2C slave and the BMC I^2C port is configured as a master. This is a point-to-point link between the BMC and the processor for BMC-initiated requests. In 2P systems, the I2C_3 signal of each socket connects to the same BMC I2C bus. The processor I2C_3 SMBALERT (ALERT_L) output is used as the SMB_ALERT_L signal to alert the BMC of critical events or errors. In 2P systems, the I2C_3 alert signal for each socket connects to the BMC. The decision to OR-tie the signals is up to the system implementer.
- The processor UART0 should pipe processor console output to the BMC to support Serial Over LAN (SOL) functionality with a remote station.
- A processor General-Purpose Input/Output (GPIO) output (CPU_FW_BOOT_OK), acting as a Software Ready signal, should connect to an input GPIO of the BMC. This informs the BMC that the processor is ready to communicate with the BMC. In a 2P system, both socket output signals connect to the BMC.

  Note: It is not recommended to OR-tie these signals.

- The non-secure (NS) GPIO23 input signal should connect to an output GPIO (BMC_CPU_SHD_REQ_L) of the BMC, which functions as a graceful shutdown request signal from the BMC. Upon receiving this signal, the processor performs a graceful shutdown. This signal is needed for the Master socket only in 2P systems.
- A processor GPIO9 output (CPU_BMC_SHD_ACK_L) connects to a GPIO interrupt input of the BMC. This informs the BMC that the processor completed a graceful shutdown, previously triggered either by the BMC using the NS GPIO23 input, or by software. The BMC in turn should turn off the PSU upon receipt of this signal. This signal is needed for Master socket only when 2P is deployed.
- A processor GPIO10 output (CPU_REBOOT_ACK_L) connects to a GPIO interrupt input of the BMC to inform the BMC that a processor reboot executed from the Operating System (OS). This signal is required for the master socket only in 2P systems.
- The processor SYS_RESET_L signal should connect to the BMC so that the BMC can reset the processor. In a 2P system, both socket signals connect to the BMC.

  Note: It is not recommended to OR-tie these signals.

- The CPU_BMC_OVERTEMP_L (also known as External OVERTEMP) output signal from the processor must connect to the BMC to alert the BMC of critical temperature conditions requiring power shutdown. In 2P systems, both socket output signals connect to the BMC. The decision to OR-tie the signals is up to the system implementer.
- The CPU_BMC_HIGHTEMP_L signal of the processor (also known as *External HIGHTEMP*) is bidirectional and by default is an input. The CPU drives this signal as an output only when firmware detects a high temperature condition. The BMC uses this signal, which must connect to the BMC GPIO output, to notify the processor of HIGHTEMP conditions observed by the BMC. The SoC VRD and Cluster Processor Module (CPM) VRD VRHOT_L signals also drive the CPU_BMC_HIGHTEMP_L signal. When the processor detects external HIGHTEMP signals, the processor reduces CPU frequencies until the signals clear. In 2P systems, both socket signals connect to the BMC.

  Note: It is not recommended to OR-tie these signals.

[1] The ALERT_L mnemonic is used generically to denote the I2C SMBALERT# input or output pins.

- A CPU_FAULT_ALERT signal from the processor should connect to a BMC GPIO interrupt input to inform the BMC that the processor encountered a fault or unrecoverable error while booting before SCP_BOOT_OK is asserted. In 2P systems, a CPU_FAULT_ALERT signal is required per socket.
- A PMIN (GPIO12) signal connects to a BMC output that functions as the power management minimum power control request signal. Upon receiving a BMC request, the processor immediately throttles to the lowest frequency/voltage. In 2P systems, a PMIN signal is required per socket.

2.2 GPIO Table

Table 1 provides more detailed descriptions of the signals described in the preceding section.

Table 1: GPIO Assignments

MNEMONICS	SIGNAL	DIR FROM PROCESSOR	SOCKET0	SOCKET1	COMMENTS
CPU_FW_BOOT_OK	GPIO8	OUT	Yes	Yes	Set by the host to inform the BMC if the host is in ready status; HIGH if the host is in ready status.
BMC_CPU_SHD_REQ_L	GPIO23	IN	Yes	N/A	Input to the host from the BMC to request a graceful shutdown; LOW level triggered.
CPU_SHD_ACK_L	GPIO9	OUT	Yes	N/A	Output from the host to BMC. Asserted LOW to acknowledge a shutdown request from the BMC. The processor also asserts this when it finishes a soft shutdown request from the OS.
CPU_REBOOT_ACK_L	GPIO10	OUT	Yes	N/A	Output from the host to the BMC. Asserted LOW to notify the BMC that a software reset executed from the OS.
CPU_BMC_OVERTEMP_L	OVERTEMP	OUT	Yes	Yes	Output LOW from the host to the BMC to indicate an OVERTEMP event. The OVERTEMP event initiates a power-off sequence for the entire processor.
BMC_CPU_RST_L	SYS_RESET	IN	Yes	Yes	Input to the host from the BMC or Reset button. Asserted LOW to reset the host.
I2C4_ALERT_L	ALERT	IN	Yes	N/A	LOW level triggered from BMC to host to notify the host of an event on the SSIF interface.
CPU_BMC_HIGHTEMP_L	HIGHTEMP	Bi-directional	Yes	Yes	At boot, this is configured as in input. At internal high temperature, this is configured as an output to BMC. On BMC detection of high temperature, assert by BMC.

MNEMONICS	SIGNAL	DIR FROM PROCESSOR	SOCKET0	SOCKET1	COMMENTS
I2C3_ALERT_L	ALERT	OUT	Yes	Yes	Output from the host to BMC to notify the BMC of an event/error on the I²C slave bus.
PMIN	GPIO12	IN	Yes	Yes	The BMC drives the signal HIGH to trigger host throttle to the lowest frequency/voltage
CPU_FAULT_ALERT	S-GPIO	OUT	Yes	Yes	High level triggered from the host to notify the BMC that the CPU has a fault/non-recoverable error.

2.3 Other Design Considerations

- The boot EEPROM must connect to an I²C bus configured as master on the processor side, preferably on its own bus. In 2P systems, both I²C buses connect to the same boot EEPROM. Software ensures access control.
- The BMC may have board-specific requirements to access the boot EEPROM for firmware upgrades. If so, additional circuitry may be needed to enable the boot EEPROM to connect to a BMC master I²C interface. This can be done using an I²C mux or I²C bus isolators. The processor I2C1 supports multi-master operations, so there is no need for an I2C mux between the I²C buses of the two sockets to access the boot EEPROM.
- The BMC must connect directly to the I2C_X bus and not through any I²C mux or expander.
- Slave devices accessible to and controlled directly by the BMC connect to a BMC I²C bus configured as master. Such devices include the fan controller, Field Replaceable Unit (FRU) EEPROM, ambient temperature sensor, and additional on-board thermal and power sensors.
- Slave devices accessible to and controlled directly by the processor connect to a processor master I²C bus. These devices include, but are not limited to, the Enhanced Small Form-Factor Pluggable (SFP+) modules, Real-Time Clock (RTC), GPIO expander, and so on.
- The RTC, if present, must connect to a processor master I2C_6 bus.
- The LED controller, if present, must connect to a processor master I²C bus under the control of the System Control Processor (SCP).
- The $\overline{\text{EVENT}}$ output signal from SPD DDR EEPROM connects to an ALERT_L input of the processor to notify the processor of any critical DDR temperature events.
- The $\overline{\text{ALERT}}$ output signals from the SoC VRD, CPM VRD, and DDR VRD(s) should connect to the $\overline{\text{PMALERT}}$ input signal to alert the processor of critical power conditions.
- The $\overline{\text{VRHOT}}$ output signals from the DDR VRD(s) also connect to the $\overline{\text{PMALERT}}$ signal.
- A BMC_OK signal from the BMC connects to an input GPI0 of the processor. The BMC triggers this to notify the processor that the BMC is ready to receive messages and requests on SSIF.
- A CPU_SLAVE_PRESENT_L signal from hardware circuitry connects to the BMC and processor Master Socket GPIO inputs. This signal indicates that a Host Slave Socket is present.
- SCP_AUTH_FAILURE (GPIO15) is an output signal from both sockets in a 2P system to the BMC. The signal notifies the BMC when a SCP firmware authentication fails (SMpro or PMpro images). The BMC, upon receipt of this notification, may switch to a failover EEPROM and reset the system. Note that ROM boot failures of the SCP only have the fault LED asserted (or blink) in a specific pattern.
- HOST_AUTH_FAILURE_L (GPIO7) is an output signal from master sockets in a 2P system to the BMC. The signal notifies the BMC when ATF BL1, BL2, BL31, BL32, UEFI BL33 authentication fails or when host failsafe procedure fails.

  Note: HOST_AUTH_FAILURE_L (GPIO7) is also asserted when BL33 certification is not included in the SPI-NOR image even when BL33 authentication is not enabled. To avoid this condition, ensure that a BL33 certification is included. In such cases, only an existing certification is needed to avoid this.

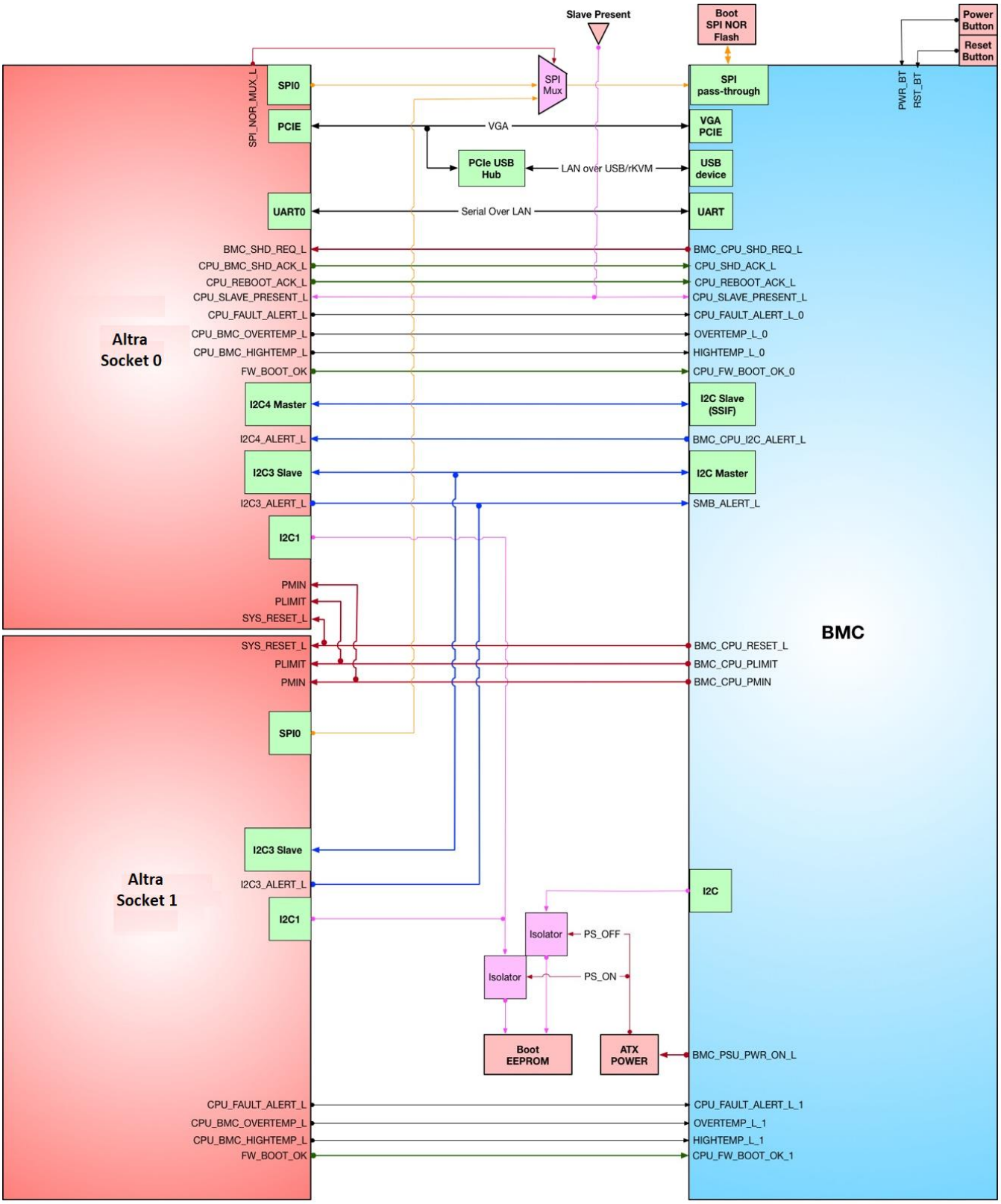
Table 2 summarizes the alert signals and additional signals not listed in *Table 1*.

Table 2: Alert and Additional Miscellaneous Signals

MNEMONIC	SIGNAL	DIR FROM PROCESSOR	SOCKET0	SOCKET1	COMMENTS
$\overline{\text{PMALERT}}$	PMALERT	IN	Yes	Yes	LOW level triggered to notify the host about events from I^2C device(s) connected to the I2C1 bus, such as DIMM high temperature.
$\overline{\text{ALERTx\_N}}$	ALERT	IN	Yes	Yes	Alert signal input from DIMM $\overline{\text{EVENT}}$ pin(s) that indicate that the DIMM temperature exceeds the warning threshold.
BMC_OK	GPI 0	IN	Yes	N/A	The BMC triggers HIGH level to notify the processor that the BMC is ready to receive SSIF messages.
MASTER_2P	GPIO	IN	Yes	Yes	On the master, the signal is asserted. On the slave, it is deasserted. **Note**: The master socket must also be routed to the BMC.
SLAVE_PRESENT_L	GPIO	IN	Yes	No	–
SLAVE_PRESENT_L	GPIO	OUT	No	Yes	The signal must be routed to the BMC to indicate the presence of the slave socket to the BMC.
SCP_AUTH_FAILURE	GPI15	OUT	Yes	Yes	SCP firmware authentication failure.
HOST_AUTH_FAILURE_L	GPI7	OUT	Yes	No	LOW level triggered to notify the host about BL1, BL2, BL31, BL32, BL33 authentication failure, or failsafe procedure failure. **Note:** HOST_AUTH_FAILURE_L (GPIO7) is also asserted when BL33 certification is not included in the SPI-NOR image even when BL33 authentication is not enabled. To avoid this condition, ensure that a BL33 certification is included. In such cases, only an existing certification is needed to avoid this.

Figure 1 illustrates an example of the processor to BMC hardware connectivity.

Figure 1: Processor-to-BMC Hardware Connectivity



3. Processor to BMC Communication

3.1 System Management Bus (SMBus)

The SMBus is a two-wire interface that various system components use to communicate with each other.

As described previously, the processor must connect to the BMC using the processor I^2C slave interface on the SMBus. The BMC must always be an I^2C master and the S-GPIO must always be an I^2C slave on the SMBus. To the BMC, the processor appears as an I^2C slave device used to query thermal sensors and VRD telemetry data such as VRD output power, VRD temperature, or sensors/system event configuration supported by the processor software. Because the BMC is the I^2C master and the processor is the slave, the BMC always issues I^2C master read requests and the processor responds to those requests in the same read transaction.

The processor must connect to another I^2C interface of the BMC using another I^2C bus. On this bus, the processor is the I^2C master, and the BMC is the I^2C slave. To the processor, the BMC appears as an I^2C slave device on this I^2C bus. This connectivity is used for host-initiated communications with the BMC.

- The SMBus standard specifies an optional signal, $\overline{\text{SMBALERT}}$, which provides an interrupt line for devices that want to trade their ability to master for a pin. $\overline{\text{SMBALERT}}$ is an active-low wired-OR signal like the $\overline{\text{SMBCLK}}$ and the $\overline{\text{SMBDAT}}$ signals. $\overline{\text{SMBALERT}}$ is used with the SMBus General Call Address. When the processor needs to communicate with the BMC, the processor asserts the $\overline{\text{SMBALERT}}$ interrupt to the BMC.
- When one BMC controls both processors in a 2P system, each processor has only one interrupt and the multiple interrupt lines are wire-ORed to the BMC for $\overline{\text{SMBALERT}}$. The BMC processes the interrupt and simultaneously accesses all $\overline{\text{SMBALERT}}$ devices using the Alert Response Address (ARA). Only devices that pull $\overline{\text{SMBALERT}}$ low acknowledge the ARA. In the one-device-per-interrupt case, the same processor always acknowledges the ARA.
- In a 2P configuration, the I2C_3 bus of each processor must connect to the BMC. Each processor independently provides sensor information to the BMC. While the master I2C_4 connects only to the BMC, the host-initiated request is driven by UEFI and the OS. At that stage of the boot process, the system is considered a uniform system and a single connection is enough.

3.2 Out-of-Band Communication

The processor does not support Out-of-Band (OOB) communication; out-of-band communication exists only for the BMC. As a reference, ARM specifies that OOB communication must use Redfish API. A remote management station connects to the BMC using the BMC OOB interfaces, such as the LAN port, and issues Redfish API to the BMC. On the processor platform, physical on-board BMC access is through the BMC management Ethernet. For example, the network manager could run remote management software or Redfish API tools on a workstation and send a Redfish API to a specific BMC. The remote software opens a network connection directly to the BMC and works on the standard Redfish API for the given commands.

3.3 In-Band Communication

As required by the Server Base Management Guide (SBMG) specification for level M2, these interfaces are required:

- Redfish
- Intelligent Platform Management Interface (IPMI)

For Redfish support, the physical interface between the host and the BMC is an Ethernet type interface. On the processor platform, this is achieved using PCIe USB Ethernet. The processor supports the Redfish protocol (0x4h) with SMBIOS Type 42. Refer to *Redfish Host Interface Specification* at https://www.dmtf.org/sites/default/files/standards/documents/DSP0270_1.0.0.pdf for details.

For IPMI, in-band communication is achieved using SSIF over I^2C or SMBus. IPMI defines standard system interfaces that system software can use to pass IPMI messages to the BMC. These interfaces are Keyboard Controller Style (KCS), System Management Interface Chip (SMIC), Block Transfer (BT), and SSIF. Over I^2C or SMBus, the processor supports only SSIF. Applications must communicate with the BMC using the `/dev/ipmi0` SSIF interface provided by the processor Linux kernel.

Detailed information about in-band communication is beyond the scope of this document.

3.4 Host-to-BMC Communication

Communication between host applications and the BMC uses IPMI over the SSIF `/dev/ipmi0` interface using the BMC slave address.

4. Detailed Software Functions

4.1 BMC-to-Processor Communication

Communication between BMC and the processor uses the processor Register Map. To query sensor data and status, the BMC can read the processor Register Map using the BMC I^2C master interface connected to the processor I^2C slave.

4.2 Error Detecting and Reporting

The processor triggers SMBALERT to notify the BMC of critical or catastrophic errors. These errors and events trigger an alert to the BMC:

- System events such as platform booting/reset and so on.
- Thermal and power events
- Core errors
- Memory errors
- PCIe errors
- Other errors
- Advanced Configuration and Power Interface (ACPI) state changes

Upon receiving the alert, the BMC must access the Register Map General-Purpose Interrupt (GPI) to identify the error source and then read the associated status information to determine the error details.

4.3 Fail-Safe Feature

During the boot process, the system may not boot correctly when there are hardware issues, such as:

- Incorrect voltage for core/SoC/DRAM (changed manually).
- Incorrect DRAM parameters (speed/interleaving mode, and so on) resulting in incorrect configuration.

These affect the processor boot process without any scope for recovery. To mitigate such situations, software implements a fail-safe[2] feature in which the system boots itself with default settings after a certain number of boot failures.

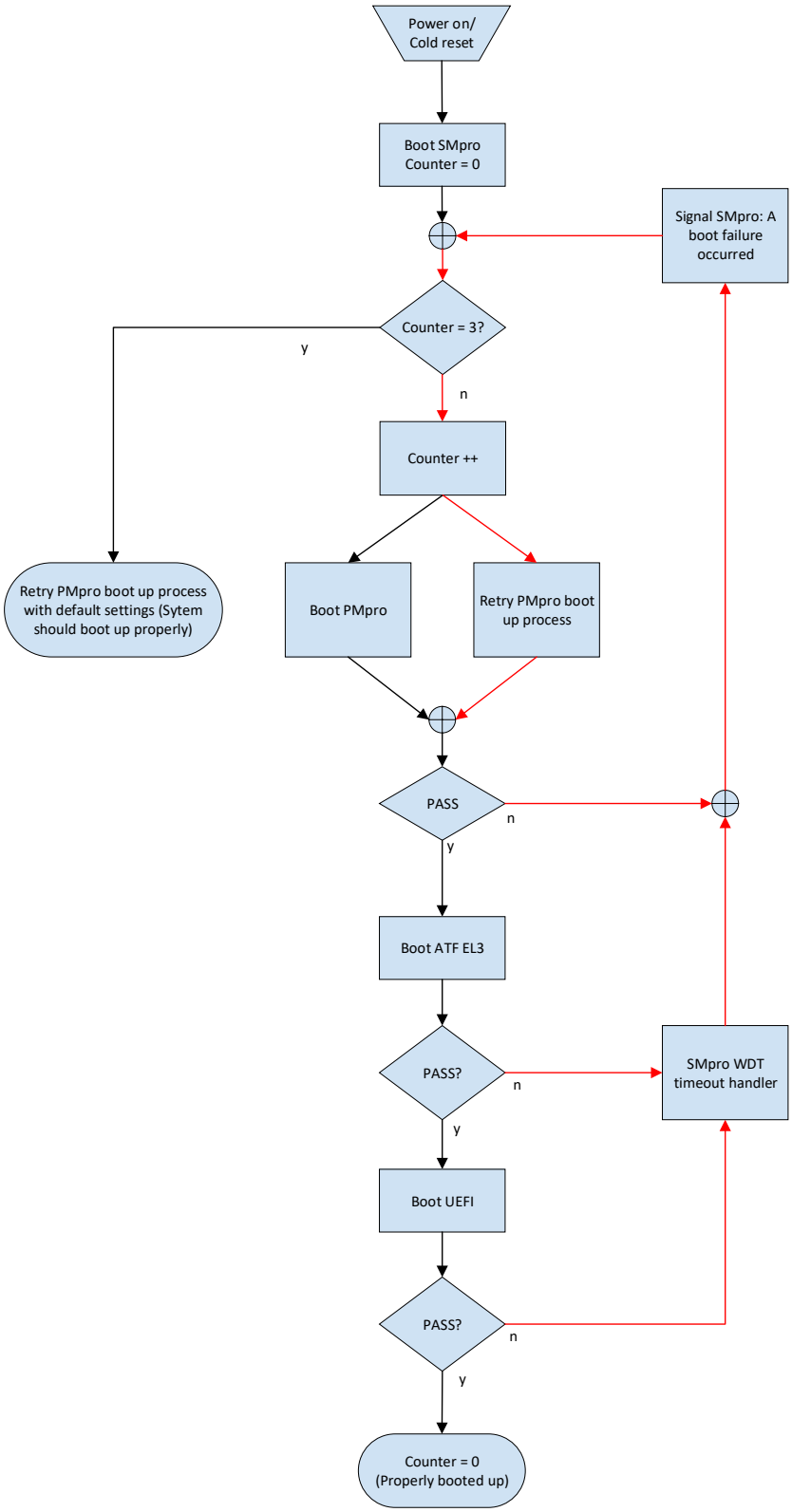
The fail-safe feature is implemented on SMpro, which always boots. SMpro maintains a counter that is initialized to 0 upon either power-on or cold reset. After SMpro boots properly and is ready to boot PMpro, the counter is set to 1. If the system boots in a clean manner to a stage which is considered good (and which is defined at the end of the UEFI BIOS stage in the boot process), the counter is cleared, indicating that the system booted successfully. If the boot failure reaches its maximum set limit, SMpro restarts the boot process in fail-safe mode. This fail-safe boot mode state is propagated to the software components in the boot process: PMpro, ATF, and UEFI.

The boot failure limit is stored in non-volatile storage. By default, the retry count is 3.

[2] Fail-safe is a BMC feature for Altra family processors.

Figure 2 illustrates the fail-safe operation sequence.

Figure 2: Fail-Safe Operation Sequence



The BMC software is expected to handle and log such information.

5. Processor Data Information Specification

The processor provides various information to the BMC through the I²C register map interface described in this section.

5.1 Extended LM75 Format

The command/address (offset) format and access mechanism from the BMC to I²C registers as provided by the processor register map is based on the National Semiconductor LM75 format.

However, the LM75 format imposes two limitations:

- The specification of the Pointer Register (to select which registers to read/write) provides only two bits for register selection, limiting the number of accessible registers to four.
- The returned value for a temperature is only one byte, enabling temperatures to be reported only in the range of 0–125°C.

To accommodate a wider range of logical functions and customer requirements, the processor I²C register map is extended to remove these restrictions:

- The Pointer Register can use up to eight bits, supporting the selection of up to 256 registers.
- Data values use two bytes instead of only one.

  Note: In some cases, data values may go up to 48 bytes.

These are the register access types:

- R Read only
- W Write only
- R/W Read and write
- W1C Write 1 to clear

5.2 Identification Definitions

Table 3 summarizes the processor register identification definitions.

Table 3: Processor Register Identification Definitions

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0x0	Register Specification Version	0x5	R	0	Major version of this specification (in hexadecimal). For example, if the major version of this specification is 5.0, this value is 0x05.
		0x1	R	1	Minor version of this specification (in hexadecimal). For example, if the minor version of this specification is 4510, then this value is 0x0A.
0x1	SCP Version	–	R	0	Major Firmware version (in hexadecimal) of the SCP firmware.
		–	R	1	Minor Firmware version (in hexadecimal) of the SCP firmware.
0x9	SCP Build ID (lower)	–	R	0	Build firmware build ID. Byte 0 of firmware build ID: **Note:** The offset is non-sequential.
		–	R	1	Byte 1 of firmware build ID.

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0xA	SCP Build ID (upper)	–	R	0	Byte 2 of the firmware build ID.
		0x00	R	1	Byte 3 of the firmware build ID.
0x2	Manufacturer ID	0x3A	R	0	Manufacturer ID, LSB first, binary encoded. This is the Internet Assigned Numbers Authority (IANA) Private Enterprise ID following the IPMI specification for Manufacturing ID.
		0xCD	R	1	–
0x3	Device ID	0x01	R	0	Device ID and revision. 0x01 for Altra.
		0x02	R	0	Device ID and revision. 0x02 for Altra Max.
		0xA0	R	1	0xA0 for revision A0. 0xA1 for revision A1. 0xB0 for revision B0. 0xB1 for revision B1 and so on.

5.3 Capability Register Definitions

The Capability Registers provide various capability information about the underlying hardware and firmware. *Table 4* describes these registers in detail.

Table 4: Capability Register Definitions

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0x5	Analog Sensor Support	–	R	0	Analog Sensor Support: 0: Reserved. 1: SoC VR Temp. 2: DIMM VR Temp. 3: Core VR Temp. 4: DIMM Temp. 5: RCA VR Temp. 6..7: Reserved.
		–	R	1	0..7: Reserved.
0x6	Analog Power Sensor Support	–	R	0	0: DIMM VR1 Power. 1: DIMM VR2 Power. 2: Core VR_Power. 3: SoC_VR_Power. 4: RCA VR Power 5..7: Reserved.
		–	R	1	Reserved.

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0x7	Analog Voltage Sensor Support	–	R	0	0: DIMM VR1 Voltage. 1: DIMM VR2 Voltage. 2: Core VR Voltage. 3: SoC VR Voltage. 4: RCA VR Voltage. 5..7: Reserved.
	Analog Current Sensor Support	–	R	1	0: DIMM VR1 Current. 1: DIMM VR2 Current. 2: Core VR Current. 3: SoC VR Current. 4: RCA VR Current. 5..7: Reserved.
0x8	Other Capabilities	–	R	0	0: Reserved. 1: ACPI Collaborative Processor Performance Control (CPPC) support. 2. ACPI Power limit control support. 3..7: Reserved.
		–	R	1	Reserved.
0xB	Core Cluster Count	–	R	0	The number of dual-core clusters in the processor; Altra has up to 40 core clusters and Altra Max has up to 64. **Note:** The offset is non-sequential.
		–	R	1	Reserved
0xC	System Cache/ PCIe Count	–	R	0	Number of PCIe controllers. This is the number of PCIe controllers in the platform.
		–	R	1	Number of System Level Caches (SLCs). The number of SLCs instantiated in the platform.
0xD	Socket Info	0x1	R	0	0: Socket0 presence. 1: Socket1 presence.
		0x0	R	1	Reserved.
0xE	Socket TDP	–	R	0..1	Socket TDP in watts.
0xF	Socket TM1	–	R	0..1	Socket TM1/T_{CJ} in °C.

5.4 Logical Power Sensor Definitions

Processor firmware provides a framework for accessing sensors, using a functional model in which board-specific firmware reports logical power sensor data.

The sensor data format follows:

- 0xFFFF – This sensor data is either missing or is not supported by the device.

Check the listed associated Analog Sensor Support registers to determine whether a sensor is supported (analog sensor support bit is set to 1). If the sensor is supported, 0xFFFF value of the corresponding sensor register indicates whether the value is valid (0) or invalid (1).

A sensor is invalid if:

- It is not supported as indicated by the Analog Sensor Support register bit.
- It is not present (such as a missing DIMM) or if the sensor is not supported by the (replaceable) device (such as lack of support by the DIMM type for temperature reading). Reading this sensor returns the value 0xFFFF.

Table 5 summarizes logical sensor definitions.

Table 5: Logical Power Sensor Register Definitions

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0x10	SoC Temperature	–	R	0	9-bit temperature in °C, ranging from -255 to +256. Byte0: LSB of the temperature. Bit 0-7: Temperature.
		–	R	1	Byte1: MSB of the temperature. Bit 0: Temperature. Bit 1-7: Reserved.
0x11	SoC VRD Temp	–	R	0..1	Highest temperature reported by the SoC VRDs: Same format as SoC Temperature. **Note**: SoC VRD corresponds to the +0V8_VDDC_SOC_S0 of the Mt Jade reference design.
0x12	DIMM VRD Temp	–	R	0..1	Highest temperature reported by both DIMM VRDs: Same format as SoC Temperature. **Note**: DIMM VRDs correspond to the +1V2_VDDQ0123_S0 and +2V2_VDDQ4567_S0 of Mt Jade reference design.
0x13	Core VRD Temp	–	R	0..1	Highest temperature reported by the Core VRDs: Same format as SoC Temperature. **Note**: Core VRD corresponds to the +0V75_PCP_S0 of Mt Jade reference design.
0x14	CH0 DIMM0 Temp	–	R	0..1	Temperature of DIMM0 on CH0: Same format as SoC Temperature.
0x15	CH1 DIMM0 Temp	–	R	0..1	Temperature of DIMM0 on CH1: Same format as SoC Temperature.
0x16	CH2 DIMM0 Temp	–	R	0..1	Temperature of DIMM0 on CH2: Same format as SoC Temperature.
0x17	CH3 DIMM0 Temp	–	R	0..1	Temperature of DIMM0 on CH3: Same format as SoC Temperature.
0x18	CH4 DIMM0 Temp	–	R	0..1	Temperature of DIMM0 on CH4: Same format as SoC Temperature.
0x19	CH5 DIMM0 Temp	–	R	0..1	Temperature of DIMM0 on CH5: Same format as SoC Temperature.
0x1A	CH6 DIMM0 Temp	–	R	0..1	Temperature of DIMM0 on CH6: Same format as SoC Temperature.
0x1B	CH7 DIMM0 Temp	–	R	0..1	Temperature of DIMM0 on CH7: Same format as SoC Temperature.

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0x1C	RCA VRD Temp.	–	R	0..1	Highest temperature reported by the RCA VRD. Same format as SoC temperature.
0x1D – 0x1F	–	–	–	–	Reserved.
0x20	Core VRD Power	–	R	0..1	The average power in 5 seconds 10-bit power consumption in watts range from 0 to 1023. Byte 0: (LSB). 0..7: Power Consumption. Byte 1: (MSB). 0..1: Power Consumption. 2..7: Reserved.
0x21	SoC IO Power	–	R	0..1	This is the SoC IO power which includes the SoC, PCIe, DDR, PHY powers (Non-Core power) Same format as Core VRD Power.
0x22	DIMM VRD1 Power	–	R	0..1	Same format as Core VRD Power.
0x23	DIMM VRD2 Power	–	R	0..1	Same format as Core VRD Power. Power as reported by the second DIMM VRD (if available).
0x24 – 0x25	–	–	–	–	Reserved.
0x26	Core VRD Power mW	–	R	0..1	The average power in 5 seconds Same format as Core VRD Power. This is the mW portion (remainder) of Core VRD Power. The total Core VRD Power is calculated as sum of registers 0x20 and 0x26.
0x27	SoC IO Power mW	–	R	0..1	Same format as Core VRD Power. This is the mW portion (remainder) of SoC IO Power. The total SoC VRD Power is calculated as the sum of registers 0x21 and 0x27.
0x28	DIMM VRD1 Power mW	–	R	0..1	Same format as Core VRD Power. This is the mW portion (remainder) of DIMM VRD1 Power. The total DIMM VRD1 Power is calculated as the sum of registers 0x22 and 0x28.
0x29	DIMM VRD2 Power mW	–	R	0..1	Same format as Core VRD Power. This is the mW portion (remainder) of DIMM VRD2 Power. The total DIMM VRD2 Power is calculated as the sum of 0x23 and 0x29.
0x2A	RCA VRD Power	–	R	0..1	10-bit power consumption in watts range from 0 to 1023. Byte 0: (LSB). 0..7: Power Consumption. Byte 1: (MSB). 0..1: Power Consumption. 2..7: Reserved.

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0x2B	Reserved	–	R	0..1	Reserved.
0x2C – 0x31	–	–	–	–	Reserved.
0x32	MEM HOT Threshold	–	R/W	0..1	Same format as SoC Temperature. MEM HOT Threshold is the value at which a DIMM triggers its MEMHOT event when its temperature exceeds the threshold.
0x33	SoC VR HOT Threshold	–	R/W	0..1	Same format as SoC Temperature.
0x34	Core VRD Voltage	–	R	0..1	Core voltage: 15-bit voltage in mV.
0x35	SoC VRD Voltage	–	R	0..1	SoC voltage: 15-bit voltage in mV.
0x36	DIMM VRD1 Voltage	–	R	0..1	DIMM VRD1 voltage: 15-bit voltage in mV.
0x37	DIMM VRD2 Voltage	–	R	0..1	DIMM VRD2 voltage: 15-bit voltage in mV.
0x38	RCA VRD Voltage	–	R/W	0..1	RCA VRD voltage: 15-bit voltage in mV.
0x39	Core VRD Current	–	R	0..1	Core Current: 15-bit current in mA.
0x3A	SoC VRD Current	–	R	0..1	SoC Current: 15-bit current in mA.
0x3B	DIMM VRD1 Current	–	R	0..1	DIMM VRD1 current: 15-bit current in mA.
0x3C	DIMM VRD2 Current	–	R	0..1	DIMM VRD2 current: 15-bit current in mA.
0x3D	RCA VRD Current	–	R	0..1	RCA VRD current: 15-bit current in mA.
0x3E – 0x3F	–	–	–	–	Reserved.
0x40	CH0 DIMM1 Temp	–	R	0..1	Temperature of DIMM1 on CH0: Same format as SoC Temperature.
0x41	CH1 DIMM1 Temp	–	R	0..1	Temperature of DIMM1 on CH1: Same format as SoC Temperature.
0x42	CH2 DIMM1 Temp	–	R	0..1	Temperature of DIMM1 on CH2: Same format as SoC Temperature.
0x43	CH3 DIMM1 Temp	–	R	0..1	Temperature of DIMM1 on CH3: Same format as SoC Temperature.

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0x44	CH4 DIMM1 Temp	–	R	0..1	Temperature of DIMM1 on CH4: Same format as SoC Temperature.
0x45	CH5 DIMM1 Temp	–	R	0..1	Temperature of DIMM1 on CH5: Same format as SoC Temperature.
0x46	CH6 DIMM1 Temp	–	R	0..1	Temperature of DIMM1 on CH6: Same format as SoC Temperature.
0x47	CH7 DIMM1 Temp	–	R	0..1	Temperature of DIMM1 on CH7: Same format as SoC Temperature.
0x48 – 0x4F	–	–	–	–	Reserved.

5.5 GPI Mask Register Definitions

These registers are used to control (enable or disable) different alert sources. Each bit indicates whether a status is enabled or disabled.

These registers are initialized to defaults at boot time and can be updated by BMC. The BMC can set a bit in this register to one to enable, or to zero to disable the reporting of such an alert. If an alert source is not supported, the corresponding bit is shown as disabled, and any attempt to enable it has no effect.

- 1: Disabled the alert of the corresponding source
- 0: Enabled the alert of corresponding source

Table 6 describes the GPI Mask registers in detail.

Note: A value (0xFFFF) is returned when reading any GPI registers (mask/status/source registers) is invalid value. BMC should consider ignoring this transaction and restart it.

Table 6: GPI Mask Register Definitions

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0x50	GPI Control #0	0x80	R/W	0	0..2: Reserved. 3: Platform Booting. 4: Critical Stop. 5..7: Reserved.
		–	–	1	Reserved
0x51	GPI Control #1	0x07	R/W	0	0: SoC VR HOT/Warn/Fault. 1: Core VR HOT/Warn/Fault. 2: DIMM VRD HOT/Warn/Fault. 3..7: Reserved.
		–	–	1	Reserved.
0x52	GPI Control #2	0x01	R/W	0	0: DIMM HOT. 1: NVDIMM-N Event. 2: Refresh Rate Event. 3..7: Reserved
		–	–	1	Reserved.

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0x53	GPI Control #3	0xEF	R/W	0	0: Core Errors. 1: Memory Errors. 2: Reserved. 3: PCIe Errors. 4: Reserved. 5: Other SoC Errors. 6: ACPI state change. 7: Boot Errors.
		0x01	R/W	1	0: RAS internal error. 1..7: Reserved.
0x54	GPI CE/UE Mask	0x45	R/W	0	GPI Uncorrectable Error (UE)/CE mask: 0: Core CE. 1: Core UE. 2: DIMM CE. 3: DIMM UE. 4: Reserved. 5: Reserved. 6: PCIe CE. 7: PCIe UE. A 1 bit indicates a mask of an error type. Masking an error prevents GPI for the alert and updating the status registers.
		0x01	R/W	1	0: Other. 1..7: Reserved. A 1 bit indicates a mask of an error type.
0x55 – 0x5F	–	–	–	–	Reserved.

5.6 GPI Source Registers

Each bit in the GPI Source registers, summarized in *Table 7*, denotes the presence or absence of a specific alert source. A value of 1 indicates that an ALERT is present. If an alert source is not supported, it appears as 0. If any alert source is present, the SMBALERT signal is triggered to the BMC to notify the alert. When the BMC clears all alert sources, the processor deactivates the SMBALERT signal and continues to update alert sources.

Upon receiving an SMBALERT signal from the processor, the BMC must first read the GPI Data Set register to determine which GPI Data Set (Data Set #0/Data Set #1/Data Set #2/Data Set #3) register is read next to determine the source of the alert(s). For GPI Data Set #0, each bit already indicates the associated status of the Alert source. For GPI Data Set #1/#2/#3, the BMC must read the corresponding GPI Status registers to find more details about the alerts.

Table 7: GPI Source Register Definitions

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0x60	GPI Data Set	0x0	R	0	Each bit in this register denotes the presence of one or more alerts in the Global Alert data sets: 0: Data set #0. 1: Data set #1. 2: Data set #2. 3: Data set #3.
				1	Reserved.
0x61	GPI Data Set #0	0x0	R	0	0..2: Reserved. 3: Platform Booting. 4: Critical Stop. 5..7: Reserved. For platform booting, see Boot stage registers. A failure that triggers the system to stop normal functionality triggers Critical Stop. An example is TPC over temperature.
				1	Reserved.
0x62	GPI Data Set #1	0x0	R	0	0: SoC VR HOT/Warn/Fault. 1: Core VR HOT/Warn/Fault. 2: DIMM VR HOT/Warn/Fault. 3..7: Reserved. **Note**: Read the VRD, Core, and DIMM VRD registers to identify which VRD controller is HOT/Warn/Fault.
				1	Reserved.
0x63	GPI Data Set #2	0x0	R	0	0: DIMM HOT (1*). 1: NVDIMM-N Event (2*). 2: Refresh Rate Event (3*). 3..7: Reserved. **Note** (1*): Read the DIMM Hot Error to identify which DIMM channel is HOT. (2*): Read the NVDIMM-N Event Information register (0xB8) to identify the event. (3*): Read the Memory Channel Refresh Rate Status (0x96) to identify which channel is in 2X refresh rate.
				1	Reserved.

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0x64	GPI Data Set #3	0x0	R	0	0: Core Errors. 1: Memory Errors. 2: Reserved. 3: PCIe Errors. 4: PCIe Hot Plug. 5: Other Errors. 6: ACPI State Change. 7: Boot Errors.
				1	0: RAS internal error. 1-7: Reserved.
0x65 – 0x6F	–	–	–	0..1	Reserved

Table 8 provides additional information about the behavior of interrupts and lists the alerts that the BMC can clear.

Table 8: GPI Interrupt Alert Behaviors Definitions

GPI STATUS	MEANING	GPI BEHAVIOR
Platform Booting	The processor is booting.	Triggered by CPU or SoC when it starts booting.
Critical Stop	CPU halts.	Triggered by CPU when OS crashes or certain events cause the CPU to hang.
SoC VR Hot	VR for SoC is in HIGHTEMP.	Triggered by CPU when VR for SoC is in HIGHTEMP.
Core VR Hot	VR for CPM is in HIGHTEMP.	Triggered by CPU when VR for Core is in HIGHTEMP.
DIMM VRD Hot	VRD for DIMM is in HIGHTEMP.	Triggered by CPU when VRD1 for DIMM is in HIGHTEMP.
DIMM Hot	DIMM[y] at channel [x] is in HIGHTEMP.	Triggered by CPU when any DIMM at any channel is in HIGHTEMP.
Core Errors	CPM/CPU has some errors.	Triggered by CPU when Core has an error.
Memory Errors	Memory has some errors.	Triggered by CPU when Memory has an error.
System Cache Errors	System cache has some errors.	Triggered by CPU when system cache has an error.
PCIe Errors	PCIe has some errors.	Triggered by CPU when any PCIe controller has any error.
PCIe Hot Plug	PCIe hot plug has some actions (remove/insert)	Triggered by a PCIe hot plug when devices are removed from or inserted into PCIe ports
Other Errors	All other errors.	Triggered by CPU when run-time watchdog expires.

GPI STATUS	MEANING	GPI BEHAVIOR
In-band firmware upgrade	In-band firmware upgrade is executed	Triggered by CPU when any firmware component is upgraded via in-band process. BMC must read registers to determine which firmware component is upgraded.
ACPI State Change	ACPI state change.	Triggered by CPU when it has any change of ACPI S-State. BMC should read register System State to determine the target state that the system has entered.
Boot Errors	System cannot boot properly.	Triggered by CPU when system cannot boot properly. BMC should read registers to determine how many times boot failed and boot status.

5.7 GPI Status Registers

The GPI Source registers provide the alert source, while the GPI Status registers specify the alert that occurs on that source. *Table 9* summarizes the details of each register. When the BMC receives an alert notification, the BMC must read the corresponding GPI Status registers to determine more about the alert. Depending upon the alert, the BMC must handle and clear the alert.

Table 9: GPI Status Register Definitions

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0x70	Core, DIMM, SLC, PCIe, and Other errors	0x0	R	0	Bit mask: Identifies availability of Core, DIMM, SLC, PCIe, and Other errors: Bit 0: Core CE error. Bit 1: Core UE error. Bit 2: DIMM CE error. Bit 3: DIMM UE error. Bit 4: Reserved. Bit 5: Reserved. Bit 6: PCIe CE error. Bit 7: PCIe UE error
		0x0	R	1	Bit 0: Other CE error. Bit 1: Other UE error. Bit 2..7: Reserved.
0x71 – 0x77	Reserved	0x0	R/W	0..1	Reserved.

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0x78	VRD Fault/Warning Error	0x0	R/W1C	0	VRD fault/warning event: Bit 0: SoC VRD fault/warning. Bit 1: Core VRD1 fault/warning. Bit 2: Core VRD2 fault/warning. Bit 3: Core VRD3 fault/warning. Bit 4: DIMM VRD1 fault/warning. Bit 5: DIMM VRD2 fault/warning. Bit 6: DIMM VRD3 fault/warning. Bit 7: DIMM VRD3 fault/warning. A fault indicates a fault as reported by the VRD. A warning indicates a warning event as reported by the VRD. For more information, see the VRD specification for fault and warning events. Note that a fault results in VRD shutdown.
		0x0	R/W1C	1	Bit 0: DIMM VRD1 fault/warning. Bit 1: DIMM VRD2 fault/warning. Bit 2: DIMM VRD3 fault/warning. Bit 3: DIMM VRD4 fault/warning.
0x79	VRD Hot	0x0	R/W1C	0	Identify which VRD controller has error: Bit 0: SoC VRD is HOT. Bit 1..3: Reserved. Bit 4: Core VRD1 is HOT. Bit 5: Core VRD2 is HOT. Bit 6: Core VRD3 is HOT. Bit 7: Reserved.
			R/W1C	1	Bit 0: DIMM VRD1 is HOT. Bit 1: DIMM VRD2 is HOT. Bit 2: DIMM VRD3 is HOT. Bit 3: DIMM VRD4 is HOT. Bit 4: Reserved. Bit 5: Reserved. Bit 6: Reserved. Bit 7: Reserved. HOT refers to an over temperature event. Fault/warning refers to all other faults/warnings as reported by the VRD. The VRD hot bit is set when alert from VRD or threshold of temperature is crossed.
0x7A	DIMM Hot Error	0x0	R/W1C	0	Identify which DIMM channel has error: Bit 0: DIMM0 channel 0 is HOT. Bit 1: DIMM0 channel 1 is HOT. Bit 2: DIMM0 channel 2 is HOT. Bit 3: DIMM0 channel 3 is HOT. Bit 4: DIMM0 channel 4 is HOT. Bit 5: DIMM0 channel 5 is HOT. Bit 6: DIMM0 channel 6 is HOT. Bit 7: DIMM0 channel 7 is HOT. DIMM hot bit is set when alert from DIMM or threshold of temperature is crossed.

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
			R/W1C	1	Identify which DIMM channel has error: Bit 0: DIMM1 channel 0 is HOT. Bit 1: DIMM1 channel 1 is HOT. Bit 2: DIMM1 channel 2 is HOT. Bit 3: DIMM1 channel 3 is HOT. Bit 4: DIMM1 channel 4 is HOT. Bit 5: DIMM1 channel 5 is HOT. Bit 6: DIMM1 channel 6 is HOT. Bit 7: DIMM1 channel 7 is HOT. DIMM hot bit is set when alert from DIMM or threshold of temperature is crossed.
0x7B	Boot #1 Error	0x0	R/W1C	0..1	Indicates how many times the system fails to boot with the last known configuration and reverts to factory defaults. The Watchdog status is also asserted.
0x7C	Boot #2 Error	0x0	R/W1C	0..1	Indicates how many times the system fails to boot with the normal configuration and reverts to the last known setting. The Watchdog status is also asserted.
0x7D	Watchdog/Other Status	0x0	R/W1C	0	Indicates that the run-time watchdog expired: Bit 0: Non-secure WDT expired. Bit 1: Secure WDT expired. Bit 2: Firmware WDT expired. Bit 3..7: Reserved.
		0x0	R/W1C	1	0..7: Reserved.
0x7E	RAS internal error	0x0	R/W1C	0..1	Bit 0: Error from SMpro. Bit 1: Error from PMpro. Bit 2..15: Reserved.
0x7F	SPI-NOR Failover	0x0	R/W1C	0..1	Indicates these authentication failures: Bit 0: ATF BL1 fails authentication Bit 1: ATF BL2 fails authentication Bit 2: ATF BL31 fails authentication Bit 3: ATF BL32 fails authentication Bit 4: UEFI BL33 fails authentication Bit 5: Failsafe procedure fails Bit 6: ATF Slim image authentication fails Bit 7: DBB authentication fails Bit 8..15: Reserved When these failures occur, Boot Error bit in GPI Data Set #3 (0x64) is also set.

5.8 Core Error Register Definitions

Table 10 provides detailed information about core and system cache error registers.

Table 10: Core Register Definitions

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0x80	Core CE error count	0x0	R/W	0	Number of core CE errors available. Any write removes the oldest instance.
			R/W	1	Flags. Bit 0: Overflow – indicates dropped error record All other bits are reserved and must be zero. The maximum error count for this error type (both CE and UE) is 32.
0x81	Core CE error length	0x0	R	0..1	Length of core CE error record.
0x82	Core CE error data	–	R	0..47	Raw core CE error record. Usage: 1. Read the count (0x80) for total CE error. 2. For each: a. Read the length (0x81). b. Read the data (0x82). c. Write to the count (0x80) to advance to next. See *Section 5.8.1* for format details for core errors.
0x83	Core UE error count	0x0	R/W	0	Number of core UE error available. Any write removes the oldest instance.
			R/W	1	Flags. Bit 0: Overflow – indicates dropped error record All other bits are reserved and must be zero. The maximum error count for this error type (both CE and UE) is 32.
0x84	Core UE error length	0x0	R	0..1	Length of core UE error record.
0x85	Core UE error data	–	R	0..47	Raw core UE error record. Usage: 1. Read the count (0x83) for total UE errors. 2. For each: a. Read the length (0x84). b. Read the data (0x85). c. Write to the count (0x83) to advance to the next. See *Section 5.8.1* for format details for core errors.
0x86 – x8F	–	–	–	–	Reserved

5.8.1 CE/UE Error Data Record Format

The format of the CE/UE error data record definition is defined in this section. The error types are listed in *Table 11*.

Table 11: CE/UE Error Type

ERROR TYPE	HARDWARE ERROR TYPE NAME
0x00	CPM (Core error)
0x01	MCU (Memory error)
0x02	Coherent Mesh Interconnect (CMI); snoop control/System Address Map (SAM), and so on. Informally called Mesh.
0x03	2P CCIX (also called other error)
0x04	2P ALI (also called other error)
0x05	GIC (also called other error)
0x06	SMMU (also called other error)
0x07	PCIe AER
0x08	PCIe Host Bridge (HB)
0x09	OCM (also called other error)
0x0A	SMpro (also called other error)
0x0B	PMpro (also called other error)

5.8.1.1 Hardware Error Type 0x00 to 0x0B CE/UE Data Record Format

The format of hardware error types 0x00 to 0x0B CE/UE data record is listed in *Table 12*.

Table 12: Hardware Error Type 0x00 to 0x0B CE/UE Data Record Format

OFFSET	FIELD	SIZE (BYTE)	DESCRIPTION
0x00	Error Type	1	See *Table 13: Hardware Error Type Details*
0x01	Subtype	1	See *Table 13: Hardware Error Type Details*
0x02	Instance	2	See *Table 13: Hardware Error Type Details*
0x04	Error status	4	See ARM RAS specification for details
0x08	Error Address	8	See ARM RAS specification for details
0x10	Error Misc 0	8	See ARM RAS specification for details
0x18	Error Misc 1	8	See ARM RAS specification for details
0x20	Error Misc 2	8	See ARM RAS specification for details
0x28	Error Misc 3	8	See ARM RAS specification for details

Table 13: Hardware Error Type Details

ERROR TYPE NAME	ERROR TYPE	SUBTYPE	INSTANCE (BIT 15:14)	INSTANCE (BIT 13:00)
CPM DSU	0	0	Socket #	CPM #
CPM Core 0	0	1	Socket #	CPM #
CPM Core 1	0	2	Socket #	CPM #
MCU Error Record 1 (DRAM CE)	1	1	Socket #	MCU #
MCU Error Record 2 (DRAM UE)	1	2	Socket #	MCU #
MCU Error Record 3 (CHI Fault)	1	3	Socket #	MCU #
MCU Error Record 4 (SRAM CE)	1	4	Socket #	MCU #
MCU Error 5 (SRAM UE)	1	5	Socket #	MCU #
MCU Error 6 (DMC recovery)	1	6	Socket #	MCU #
MCU Link Error	1	7	Socket #	MCU #
Mesh XP	2	0	Socket #	X | (Y << 5) | (NS << 11)
Mesh HNI	2	1	Socket #	X | (Y << 5) | (NS << 11)
Mesh HNF	2	2	Socket #	X | (Y << 5) | NS << 11 | device << 12
Mesh CXG	2	4	Socket #	–
2P CCIX Error	3	0	Socket #	Link #
2P ALI Error	4	0	Socket #	Link #
GIC	5	0	Socket #	0
SMMU	6	0	Socket #	Root complex #
PCIe AER (Root Port)	7	0	Socket #	Segment #
PCIe AER (Device)	7	1	Socket #	Segment #
PCIe HB RCA	8	0	Socket #	Root complex #
PCIe HB RCB	8	1	Socket #	Root complex #
PCIe HB RASDP	8	8	Socket #	Root complex #
OCM Error 0 (ECC Fault)	9	0	Socket #	0
OCM Error 1 (Error Recovery)	9	1	Socket #	0
OCM Error 2 (Data Poisoned)	9	2	Socket #	0
SMpro Error 0 (ECC Fault)	10	0	Socket #	0
SMpro Error 1 (Error Recovery)	10	1	Socket #	0
PMpro Error 0 (ECC Fault)	11	0	Socket #	0
PMpro Error 1 (Error Recovery)	11	1	Socket #	0

5.9 Memory Error Register Definitions

Table 14 provides detailed information about memory error registers.

Table 14: Memory Error Registers

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0x90	Memory CE error count	0x0	R/W	0	Number of memory CE errors available. A write removes the oldest instance.
			R/W	1	Flags. Bit 0: Overflow – indicates dropped error record All other bits are reserved and must be zero. The maximum error count for this error type (both CE and UE) is 16.
0x91	Memory CE error length	0x0	R/W	0..1	Length of memory CE error records.
0x92	Memory CE error data	–	R	0..47	Raw memory CE error record. Usage: 1. Read the count (0x90) for total CE errors. 2. For each: a. Read the length (0x91). b. Read the data (0x92). c. Write to the count (0x90) to advance to next. See *Section 5.8.1* for format details related to memory errors.
0x93	Memory UE error count	0x0	R/W	0	Number of memory UE error available. A write removes the oldest instance.
			R/W	1	Flags. Bit 0: Overflow – indicates dropped error record All other bits are reserved and must be zero. The maximum error count for this error type (both CE and UE) is 16.
0x94	Memory UE error length	0x0	R	0..1	Length of memory UE error record.
0x95	Memory UE error data	–	R	0..47	Raw memory UE error record. Usage: 1. Read the count (0x93) for total UE errors. 2. For each: a. Read the length (0x94). b. Read the data (0x95). c. Write to the count (0x93) to advance to next. See *Section 5.8.1* for format details for memory errors.

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0x96	Memory Channel Refresh Rate Status	0x0	R	0	In a high temperature condition, the memory controller changes to a 2X refresh rate as required by JEDEC Specification. This register keeps the current refresh rate status for all memory channels: Bit 0: Channel 0 refresh rate status Bit 1: Channel 1 refresh rate status Bit 2: Channel 2 refresh rate status Bit 3: Channel 3 refresh rate status Bit 4: Channel 4 refresh rate status Bit 5: Channel 5 refresh rate status Bit 6: Channel 6 refresh rate status Bit 7: Channel 7 refresh rate status Value definition: 1: 2X refresh rate 0: Normal refresh rate **Note**: If any bit is set, the Refresh Rate Event bit (Bit-2 of GPI DataSet #2) is set to indicate one or more memory channels is in 2X refresh rate. Otherwise, the Refresh Rate Event bit is cleared if this register is zero (all memory channels are in normal refresh rate).
0x97 – 0x9F	–	–	–	–	Reserved.

5.10 RAS Internal Error Register Definitions

Table 15 provides detailed information about RAS internal error registers. These registers contain internal RAS information that can be provided to the Ampere firmware team for diagnostics if they are encountered. If an error type is active, no additional updates can occur until the error type is cleared.

Table 15: RAS Internal Error Registers

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0xA0	SMpro RAS internal error type	0x0	R/W1C	0..1	0: Warning. 1: Error. 2: Error with data. 3..15: Reserved. On W1C, the corresponding registers are cleared to 0.
0xA1	PMpro RAS internal error type	0x0	R/W1C	0..1	0: Warning. 1: Error. 2: Error with data. 3..15: Reserved. On W1C, the corresponding registers are cleared to 0.

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0xA2	SMpro RAS internal error info	0x0	R	0..1	The registers 0xA2 and 0xA3 store the 32-bit value of error information. 0xA2 stores the lower 16-bit value. 0xA3 stores the higher 16-bit value. For more information, refer to the section "SCP Firmware Error Codes" *in Altra Family System Control Processor User's Manual*.
0xA3		0x0	R	0..1	
0xA4	SMpro Extensive data of RAS internal error	0x0	R	0..1	The registers 0xA4 and 0xA5 store the 32-bit value of extensive data. 0xA4 stores the lower 16-bit value. 0xA5 stores the higher 16-bit value. For more information, refer to the section "SCP Firmware Error Codes" *in Altra Family System Control Processor User's Manual*.
0xA5		0x0	R	0..1	
0xAA	SMpro RAS internal warning error info	0x0	R	0..1	The registers 0xAA and 0xAB store the 32-bit value of warning information. 0xAA stores the lower 16-bit value. 0xAB stores the higher 16-bit value. For more information, refer to the section "SCP Firmware Error Codes" *in Altra Family System Control Processor User's Manual*.
0xAB					
0xA6	PMpro RAS internal error info	0x0	R	0..1	The registers 0xA6 and 0xA7 store 32-bit value of warning or error information. 0xA6 stores the lower 16-bit value. 0xA7 stores the higher 16-bit value.
0xA7		0x0	R	0..1	
0xA8	PMpro Extensive data of RAS internal error	0x0	R	0..1	The registers 0xA8 and 0xA9 store 32-bit value of extensive data. 0xA8 stores the lower 16-bit value. 0xA9 stores the higher 16-bit value. These two registers provide extra information that is specific to that error. For more information, refer to the section "SCP Firmware Error Codes" *in Altra Family System Control Processor User's Manual*.
0xA9		0x0	R	0..1	
0xAC	PMpro RAS internal warning error info	0x0	R	0..1	The registers 0xAC and 0xAD store 32-bit value of warning information. 0xAC stores the lower 16-bit value. 0xAD stores the higher 16-bit value. For more information, refer to the section "SCP Firmware Error Codes" *in Altra Family System Control Processor User's Manual*.
0xAD					

5.11 Boot Stage Register Definitions

Boot Stage Registers enable tracking the progress of system booting up to the point of successful handover to the kernel or in the case of LinuxBoot, the start of the so-called UEFI DXE stage. As a specific Boot Stage progresses, the system might report multiple Boot Status or Boot Progress codes associated with the Boot Stage. *Table 16* provides more detailed descriptions of these registers. See the section titled *Processor Boot Progress Codes* for the list of boot progress codes.

Table 16: Boot Stage Registers

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0xB0	Boot stage	0x0	R/W1C	0	This register provides the status of these boot stages: Boot status: 0: Not started. 1: Started. 2: Completed without error. 3: Failed. 0xff: Unsupported stage – Boot Stage 0 with Boot Status 2 and Boot Stage 7 with Boot Status 2 trigger GPI. Any Boot Stage with Boot Status 3 triggers GPI. – Boot Stage 3 and 4 report the status of DDR initialization and DIMM training. DDR initialization is a step in BL1 booting (Boot Stage 2), so the status of Boot Stage 3 and 4 is updated before the boot status of Boot Stage 2 changes to 0x2 (Completed without error) or 0x3 (Failed). The boot status of Boot Stage 4 is valid only if Boot Stage 3 completes without error (0x2) or failed (0x3) with training failure. – For Boot Stage 9, Boot Status 1 signals the start of OS loading or the UEFI Boot Device Selection (BDS) phase; Boot Status 2 signals receipt of the UEFI Exit Boot Service event; and Boot Status 3 indicates an error before that.

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
		0xFF	R/W1C	1	Boot stage value: 0: SMpro firmware booting. 1: PMpro firmware booting. 2: ATF BL1 firmware booting. 3: DDR initialization. 4: DDR training report status. 5: ATF BL2 firmware booting. 6: ATF BL31 firmware booting. 7: ATF BL32 firmware booting. 8: UEFI firmware booting. 9: OS booting. The first read on this register always return the value of the SMPro boot stage by default. To find all the executed boot stages, it is necessary to execute this procedure: • Write this register with: ○ Byte 0: the value of the recently read Boot Stage ○ Byte 1: must be 0x1 Read the register again. After this, the value of next Boot Stage is returned with the Boot Status in the next byte and Status Code/Boot Progress in subsequent registers 0xB1 and 0xB3.
0xB1	Boot stage low value	0x0	R	0	Boot status data value: “DDR initialization with failure” Boot Stage: 0: Bit set if a generic failure. 1: Bit set if configuration failure. 2: Bit set if training failure. 3: Bit set if ECC init failure. 4: Bit set if no-dimm plugged. **“**DDR training report status” Boot Stage: (*Valid only if the previous stage status is “DDR initialization with training failure”*) 0: Bit set if MCU4 Slot0 DIMM train failure 1: Bit set if MCU4 Slot1 DIMM train failure … 6: Bit set if MCU7 Slot0 DIMM train failure 7: Bit set if MCU7 Slot1 DIMM train failure “UEFI firmware boot” Boot Stage: 0..7: Byte 1 of UEFI Boot Progress information. In the case of Aptio, this contains the Checkpoint number sent from Aptio. In the case of TianoCore, this contains byte 0 of a 32-bit UEFI boot progress status code as defined by EDK2.

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
		0x0	R	1	"DDR initialization with Failure" Boot Stage: – If configuration failure, indicate the DIMM failure location: Bit 0: slot 0 of MCUx Bit 1: slot 1 of MCUx Bit 2..7: MCU number – Should be 0xFF for other boot stage failures such as ecc init, no-dimm, or generic failure reported in Byte 1. "DDR training report status" Boot Stage: (*Valid only if previous stage status is "DDR initialization with training failure"*) 0: Bit set if MCU0 Slot0 DIMM train failure 1: Bit set if MCU0 Slot1 DIMM train failure … 6: Bit set if MCU3 Slot0 DIMM train failure 7: Bit set if MCU3 Slot1 DIMM train failure "UEFI firmware boot" Boot Stage: 0..7: Byte 0 of UEFI check point information.
0xB2	Current boot stage	0x0	R	0	This register contains the value of the current/latest Boot Stage. Reading this register returns the value of the Boot Stage at the point of reading. Refer the entry for register 0xB0 in this table for a list of supported boot stage values.
		0x0	R	1	Reserved
0xB3	Boot stage upper value	0x0	R	0	"UEFI firmware boot" Boot Stage: 0..7: Byte 3 of UEFI check point information. Other Boot Stages: 0..7: N/A
		0x0	R	1	"UEFI firmware boot" Boot Stage: 0..7: Byte 2 of UEFI check point information. Other Boot Stages: 0..7: N/A
0xB4	Boot Stage Error Syndrome Selection	0x0	R/W	0	"DDR training report status" Boot Stage: (*Valid only if previous stage status is "DDR initialization with training failure"*) 0..3: DIMM slot failure selection. Write the slot ID to retrieve Error Syndrome in register 0xB5, for example: MCU3, slot 1: slot ID = 7 MCU7, slot 0: slot ID = 14

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
		0x0	R/W	1	Reserved
0xB5	Boot Stage Error Syndrome	0x0	R	0	**"DDR training report status" Boot Stage:** (*Valid only if previous stage status is "DDR initialization with training failure"*) 0:1 - Training failure type 0: N/A 1: PHY training failure 2: DIMM training failure 3: Reserved. 2:4 – Physical Rank error (MCU rank indexing) 5:7 – Training failure syndrome 0 PHY Training failure syndrome 0: 0: N/A 1: PHY Training Setup failure 2: PHY Write Leveling failure – See Training failure syndrome 1 for more information. 3: PHY Read Gate Leveling failure 4: PHY Read Leveling failure 5: PHY Software Training failure 6,7: Reserved DIMM Training failure syndrome 0: 0: N/A 1: DRAM VREFDQ Training failure 2: LRDIMM DB Training failure 3: LRDIMM DB Software Training failure Others: Reserved
		0x0	R	1	"DDR training report status" Boot Stage: (*Valid only if previous stage status is "DDR initialization with training failure"*) PHY Write Leveling failure syndrome 1: 0:3 - Slice number 4 - Upper Nibble Error Status b0: No Error b1: Found no rising edge. 5 - Lower Nibble Error Status b0: No Error b1: Found no rising edge. Others: N/A

5.12 NVDIMM-N Status Register Definitions

Table 17: NVDIMM-N Status Registers

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0xB8	NVDIMM-N Event Information	0x0	R/W1C	0	This register indicates the NVDIMM-N event status: Bit 0: RESTORE START Bit 1: RESTORE SUCCESS Bit 2: RESTORE FAILED Bit 3: OTHER OPERATION ERROR Bit 4: HEALTH ERROR Bit 5..6: Reserved Bit 7: Valid Bit (Data valid only if this bit is set) **Note**: • Any error bit set/clear from this register sets/clears GPI Data Set #2 Bit-1 (NVDIMM-N Event) for BMC notification. • If Bit-2, Bit-3, Bit-4 set, refer to the register NVDIMM-N Status (0xB9/0xBA) for detail error information. • If Bit-2 set, check NVDIMM-N Status with STATUS REQUEST 0x4 for RESTORE status. • If Bit-3 set, check NVDIMM-N Status with STATUS REQUEST 0x2, 0x5, 0x6 for READY, ERASE, ARM operation status. • If Bit-4 set, check NVDIMM-N Status with STATUS REQUEST 0x8 → 0xC for Health status.
		0x0	R	1	Reserved

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0xB9	NVDIMM-N Status Request	0x0	RW	0	NVDIMM-N Status Request: [7:4]: STATUS REQUEST 0x0: NVDIMM INSTALLED 0x1: OPERATIONAL STATUS 0x2: READY STATUS 0x3: CSAVE OPERATION STATUS 0x4: RESTORE OPERATION STATUS 0x5: ERASE OPERATION STATUS 0x6: ARM OPERATION STATUS 0x7: Reserved 0x8: MODULE_HEALTH 0x9: MODULE_HEALTH_STATUS0 0xA: MODULE_HEALTH_STATUS1 0xB: ERROR_THRESHOLD_STATUS 0xC: WARNING_THRESHOLD_STATUS 0xD..0xF: Reserved [3:0]: DIMM SELECT (DIMM 0 -> DIMM 15) Note: • Write the STATUS REQUEST and DIMM SELECT value into this register to get the status of NVDIMM-N by reading register 0xBA. • If STATUS REQUEST is 0x0 (NVDIMM INSTALLED), DIMM SELECT is a don't care value. • CSAVE OPERATION STATUS are monitored by BMC as described in NVDIMM-N Firmware Design Specification. STATUS REQUEST 0x3 simply provides additional information.
		0x0	RW	1	Reserved

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0xBA	NVDIMM-N Status	0x0	R	0	This register contains the returned NVDIMM-N Status following the STATUS REQUEST/DIMM SELECT written to the register 0xB9. • If STATUS REQUEST is 0x0 (NVDIMM INSTALLED), this register contains the NVDIMM-N mask (low byte) to indicate which NVDIMM-N is plugged in the system. • If STATUS REQUEST is 0x1 (OPERATIONAL STATUS), the status should be: 0x0: UNKNOWN 0x1: OPERATING (NVDIMM-N is operating) 0x2: DISABLED (NVDIMM-N is disabled due to operation failure) • If STATUS REQUEST is from 0x2 to 0x6 (Operation Status), the status should be: 0x0: NOT STARTED 0x1: STARTED 0x2: SUCCESS 0x3: FAILED 0x4: TIMEOUT • If STATUS REQUEST starts from 0x8 (Health Status), this register contains the NVDIMM-N health status. The register name and bit definition conform with the register definition in JESD245D specification, as follows: MODULE_HEALTH (Page #0, 0xA0) MODULE_HEALTH_STATUS0 (Page #0, xA1) MODULE_HEALTH_STATUS1 (Page #0, 0xA2) ERROR_THRESHOLD_STATUS (Page #0, 0xA5) WARNING_THRESHOLD_STATUS (Page #0, 0xA7) **Note**: If NVDIMM MODE is NON-NVDIMM (NVDIMM is working as regular DIMM), the status should be 0.
		0x0	R	1	• If STATUS REQUEST is 0x0 (NVDIMM INSTALLED), this register contains the NVDIMM-N mask (high byte) to indicate which NVDIMM-N is plugged in the system. • If STATUS REQUEST is others, this register contains the value of NVDIMM MODE: 0x0: NON-NVDIMM 0x1: NON-HASHED 0x2: HASHED

5.13 PCIe Error Register Definitions

Table 18 provides detailed information about PCIe error registers.

Table 18: PCIe Error Registers

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0xC0	PCIe CE Error Count	0x0	R/W	0	Number of PCIe CE errors available. Any write removes the oldest instance.
			R/W	1	Flags. Bit 0: Overflow – indicates dropped error record All other bits are reserved and must be zero. The maximum error count for this error type (both CE and UE) is 96.
0xC1	PCIe CE Error Length	0x0	R/W	0..1	Length of the PCIe CE error record.
0xC2	PCIe CE Error Data	–	R	0..47	Raw PCIe CE error record. Usage: 1. Read the count (0xC0) for total CE error. 2. For each: a. Read the length (0xC1). b. Read the data (0xC2). c. Write to the count (0xC0) to advance to next. See *Section 5.8.1* for format details related to PCIe HB errors.
0xC3	PCIe UE error count	0x0	R/W	0	Number of available PCIe UE errors. Any write removes the oldest instance.
			R/W	1	Flags. Bit 0: Overflow – indicates dropped error record All other bits are reserved and must be zero. The maximum error count for this error type (both CE and UE) is 96.
0xC4	PCIe UE error length	0x0	R	0..1	Length of the PCIe UE error record.
0xC5	PCIe UE error data	–	R	0..47	Raw PCIe UE error record. Usage: 1. Read the count (0xC3) for total CE error. 2. For each: a. Read the length (0xC4). b. Read the data (0xC5). c. Write to the count (0xC3) to advance to next See *Section 5.8.1* for format details related to PCIe HB errors.

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0xC6	PCIe Hot Plug Event Count	0x0	R/W	0	Number of PCIe hot plug events available. Any write removes the oldest instance.
			R/W	1	Flags. Bit 0: Overflow – indicates dropped hot plug record All other bits are reserved and must be zero. The maximum hot plug event count is 24.
0xC7	PCIe Hot Plug data	0x0	R	0..4	Info of PCIe controller which alerts the hot plug event: Bit 0..3: Segment number Bit 4..7: Bus number Bit 8..11: Device number Bit 12..15: Function Bit 16…19: Action (0: remove – 1: insert) Bit 20..31: reserved
0xC8 – 0xCF	Reserved	–	–	–	–

5.14 Other Errors

Table 19 provides detailed information about Other Error registers.

Table 19: Other Error Registers

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0xD0	Other CE error count	0x0	R/W	0	Number of other CE errors available. A write removes the oldest instance.
			R/W	1	Flags. Bit 0: Overflow – indicates dropped error record All other bits are reserved and must be zero. The maximum error count for this error type (CE and UE) is 8.
0xD1	Other CE error length	0x0	R	0..1	Length of other CE error record.
0xD2	Other CE error data	–	R	0..47	Raw other CE error record. Usage: 1. Read the count (0xD0) for total CE errors. 2. For each: a. Read the length (0xD1). b. Read the data (0xD2). c. Write to the count (0xD0) to advance to next. See *Section 5.8.1* for format details related to other errors.
0xD3 – 0xD7	Reserved	–	–	–	–

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0xD8	Other UE error count	0x0	R/W	0	Number of other UE error available. Any write removes the oldest instance.
			R/W	1	Flags. Bit 0: Overflow – indicates dropped error record All other bits are reserved and must be zero. The maximum error count for this error type (CE and UE) is 8.
0xD9	Other UE error length	0x0	R	0..1	Length of other UE error record. A write resets the offset of the data to 0.
0xDA	Other UE error data	–	R	0..47	Raw other UE error record. Usage: 1. Read the count (0xD8) for total CE errors. 2. For each: a. Read the length (0xD9). b. Read the data (0xDA). c. Write to the count (0xD8) to advance to next. See *Section 5.8.1* for format details related to other errors.
0xDB – 0xDF	Reserved	–	–	–	–

5.15 ACPI State Register Definitions

Table 20 summarizes details for the ACPI states for the system and cores on the processor. Additionally, the ACPI state of the system or individual cores can be changed using these writable registers.

Table 20: ACPI State Registers

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
0xE0	System State	–	R	0	0..7: System power state. On Read, returns the current System power state. The System power state encoding is: 0x00: S0, Running state. 0x01: S1, equals to Standby state in Linux (not supported). 0x04: S4, Hibernate (Suspend to Disk) (not supported). 0x05: S5, Power-off state. **Note:** In the power-off state, the register map is not accessible. This state is documented here for completeness and is applicable for external applications monitoring the system state.
			R/W1C	1	0: Indicates ACPI state has changed. 1..7: Reserved.
0xE1	Reserved	–	–	0	Reserved.
				1	
0xE2	Reserved	–	–	0	Reserved.

REGISTER ADDRESS	REGISTER NAME	INITIAL VALUE	ACCESS TYPE	BYTE	REGISTER DESCRIPTION
				1	
0xE3	CPPC Cluster Selection	0x00	R/W	0	The CPPC core cluster selection and core cluster CPPC registers are valid only when the bit “ACPI CPPC support” in the “Other Capabilities” register is set to 1. When “ACPI CPPC support” is set to 0, reads and write to these registers are ignored.
				1	Reserved.
0xE4	CPPC Cluster Data register	–	R/W	0	The CPPC core cluster selection and core cluster CPPC registers are valid only when the bit “ACPI CPPC support” in the “Other Capabilities” register is set to 1. When “ACPI CPPC support” is set to 0, reads and writes to these registers are ignored. A read from this register returns the current frequency in MHz of the cluster indexed by CPPC cluster Index register. A write to this register sets the frequency of the cluster indexed by CPPC cluster Index register to specify MHz. The valid frequency must be in the recommended range supported by the processor: Minimum frequency: 1000 MHz. Maximum Frequency: 2800 MHz. 0..7: CPU frequency lower byte.
				1	0..7: CPU frequency upper byte.
0xE5	Power Limit	TDP	R/W	0	Writes to this register set the desired SoC power limit (W). Reads from this register return the current SoC power limit (W). Value Range: • Minimum: 120 W • Maximum: Socket TDP power
				1	Reserved.
0xE6	Trigger Function	0x00	W	0	This register indicates the supported trigger functions, which BMC can request. Writing to each bit of this register triggers the corresponding function. Bit 0: firmware crash dump request Bit 1: NMI trigger request Bit [2..7]: reserved
				1	Reserved.
0xE6 – 0xEF	–	–	–	–	Reserved.

6. Processor Boot Progress Codes

Table 21 lists the boot progress codes for the processor.

Table 21: Processor Boot Progress Codes

MODULE	BOOT STATUS (REGISTER 0xB0 BYTE 0)	BOOT STAGE (REGISTER 0xB0 BYTE 1)	BOOT STAGE LOW (REGISTER 0xB1 BYTE 0)	BOOT STAGE LOW (REGISTER 0xB1 BYTE 1)	BOOT STAGE UPPER (REGISTER 0xB3 BYTE 0)	BOOT STAGE UPPER (REGISTER 0xB3 BYTE 1)	DESCRIPTION
SMpro	1	0	0	0	0	0	SMpro booting
	2	0	0	0	0	0	SMpro completed
	3	0	0	0	0	0	SMpro boot failed
	255	0	0	0	0	0	Information not available
PMpro	1	1	0	0	0	0	PMpro booting
	2	1	0	0	0	0	PMpro completed
	3	1	0	0	0	0	PMpro boot failed
	255	1	0	0	0	0	Information not available
ATF BL1	1	2	0	0	0	0	ATF BL1 booting
	2	2	0	0	0	0	ATF BL1 boot completed
	3	2	0	0	0	0	ATF BL1 boot failed
	255	2	0	0	0	0	Information not available
DDR Init	1	3	0	0	0	0	DDR initialization started
	2	3	0	0	0	0	DDR initialization completed

MODULE	BOOT STATUS (REGISTER 0xB0 BYTE 0)	BOOT STAGE (REGISTER 0xB0 BYTE 1)	BOOT STAGE LOW (REGISTER 0xB1 BYTE 0)	BOOT STAGE LOW (REGISTER 0xB1 BYTE 1)	BOOT STAGE UPPER (REGISTER 0xB3 BYTE 0)	BOOT STAGE UPPER (REGISTER 0xB3 BYTE 1)	DESCRIPTION
	3	3	X	Z	0	0	DDR initialization failed. Z indicates DIMM slot failed. X indicates initialization failure info.
	255	3	0	0	0	0	Information not available
DDR Training Error	1	4	0	0	0	0	DDR training progress started
	2	4	0	0	0	0	DDR training progress completed
	3	4	Z	Z	0	0	DDR training progress failed. Z indicates DIMM training failure info.
	255	4	0	0	0	0	Information not available
ATF BL2	1	5	0	0	0	0	ATF BL2 booting
	2	5	0	0	0	0	ATF BL2 completed
	3	5	0	0	0	0	ATF BL2 boot failed
	255	5	0	0	0	0	Information not available
ATF BL31	1	6	0	0	0	0	ATF BL31 booting
	2	6	0	0	0	0	ATF BL31 completed
	3	6	0	0	0	0	ATF BL31 boot failed
	255	6	0	0	0	0	Information not available

MODULE	BOOT STATUS (REGISTER 0xB0 BYTE 0)	BOOT STAGE (REGISTER 0xB0 BYTE 1)	BOOT STAGE LOW (REGISTER 0xB1 BYTE 0)	BOOT STAGE LOW (REGISTER 0xB1 BYTE 1)	BOOT STAGE UPPER (REGISTER 0xB3 BYTE 0)	BOOT STAGE UPPER (REGISTER 0xB3 BYTE 1)	DESCRIPTION
ATF BL32	1	7	0	0	0	0	ATF BL32 booting
	2	7	0	0	0	0	ATF BL32 completed
	3	7	0	0	0	0	ATF BL32 boot failed
	255	7	0	0	0	0	Information not available
ATF BL33 (UEFI)	1	8	UEFI Code (Byte 1)	UEFI Code (Byte 0)	UEFI Code (Byte 3)	UEFI Code (Byte 2)	UEFI booting. Code indicates the UEFI boot progress code. See UEFI boot progress code for details. Note that not all UEFI implementations use all 4 bytes code. Some implementations only use a single byte.
	2	8	UEFI Code (Byte 1)	UEFI Code (Byte 0)	UEFI Code (Byte 3)	UEFI Code (Byte 2)	UEFI complete. Code indicates the UEFI boot progress code. See UEFI boot progress code for details. Note that not all UEFI implementations use all 4 bytes code. Some implementations only use a single byte.
	255	8	0	0	0	0	Information not available

7. Document Revision History

ISSUE	DATE	DESCRIPTION
1.42	February 9, 2023	Updated: • *Table 1: GPIO Assignments* • *Section 2.3, Other Design Considerations* • *Table 2: Alert and Additional Miscellaneous Signals* • *Table 5: Logical Power Sensor Register Definitions* • *Section 5.5, GPI Mask Register Definitions* • *Table 9: GPI Status Register Definitions* • *Table 21: Processor Boot Progress Codes*
1.36	February 22, 2021	Updated: • Throughout this specification, references to “Altra” are replaced by “processor” because this specification now covers Altra and Altra Max processors. • The section titled Overview • The section titled Processor to BMC Hardware Connectivity • The section titled Processor to BMC Communication • The section titled Boot Stage Register Definitions • Table 3: Processor Register Identification Definitions • Table 4: Capability Register Definitions • Table 10: Core Register Definitions • Table 14: Memory Error Registers • Table 15: RAS Internal Error Registers • Table 16: Boot Stage Registers • Table 18: PCIe Error Registers • Table 19: Other Error Registers • Table 20: ACPI State Registers • Table 21: Processor Boot Progress Codes
1.34	August 24, 2021	• Updated the description for address 0x7F in *Table 9: GPI Status Register Definitions*
1.33	June 23, 2021	Updated: • *Table 10: Core Register Definitions* • *Table 11: CE/UE Error Type* • *Table 13: Hardware Error Type Details* • *Table 21: Processor Boot Progress Codes*
1.31	May 14, 2021	• Updated the minimum of “Power Limit” from 90 W to 120 W.
1.30	April 21, 2021	• Updated *Table 6: GPI Mask Register Definitions* • Updated *Table 7: GPI Source Register Definitions* • Updated *Table 8: GPI Interrupt Alert Behaviors Definitions* • Updated the section titled *Boot Stage Register Definitions* • Added the section titled *Processor Boot Progress Code*

ISSUE	DATE	DESCRIPTION
1.29	February 26, 2021	• Update VRD information in *Table 5: Logical Power Sensor Register Definitions* • Added PCIe hot plug information to *Table 8: GPI Interrupt Alert Behaviors Definitions* • Added maximum error count information to *Table 10: Core Register Definitions* • Added maximum error count information to *Table 14: Memory Error Registers* • Added PCIe hot plug information to *Table 18: PCIe Error Registers* • Added maximum error count information to *Table 19: Other Error Registers* • Added trigger function information to *Table 20: ACPI State Registers*
1.28	February 9, 2021	• Replace SoC VRD power registers by SoC IO power registers at *Table 5: Logical Power Sensor Register Definitions* • Add description for “Power Limit” Register at *Table 20: ACPI State Registers*
1.27	January 27, 2021	Minor updates to these tables: • *Table 7: GPI Source Register Definitions* • *Table 10: Core Register Definitions* • *Table 14: Memory Error Registers* • *Table 18: PCIe Error Registers* • *Table 19: Other Error Registers* • *Table 20: ACPI State Registers*
1.26	January 12, 2021	Minor updates to these tables: • *Table 20: ACPI State Registers*
1.25	December 23, 2020	Added: • *Section 5.12 (NVDIMM-N Status Register Definitions)* Minor updates in these tables: • *Table 1: GPIO Assignments* • *Table 4: Capability Register Definitions* • *Table 5: Logical Power Sensor Register Definitions* • *Table 6: GPI Mask Register Definitions* • *Table 7: GPI Source Register Definitions* • *Table 9: GPI Status Register Definitions* • *Table 10: Core Register Definitions* • *Table 14: Memory Error Registers* • *Table 16: Boot Stage Registers* • *Table 17: NVDIMM-N Status Registers* • *Table 18: PCIe Error Registers* • *Table 19: Other Error Registers*
1.22	September 20, 2020	Updated: • *Table 4: Capability Register Definitions* • *Table 5: Logical Power Sensor Register Definitions* • *Table 15: RAS Internal Error Registers*
1.21	July 16, 2020	Updated: • *Chapter 2* (*Hardware Interfaces*) • *Chapter 5* (*Processor Data Information Specification*)

ISSUE	DATE	DESCRIPTION
1.16	May 24, 2020	Updated: • *Section 2.3, Other Design Considerations* • *Section 3.1, System Management Bus (SMBus)* • *Table 6: GPI Mask Register Definitions* • *Table 9: GPI Status Register Definitions* • *Table 20: ACPI State Registers*
1.13	April 15, 2020	Updated: • *Section 3.1* (*System Management Bus (SMBus)*) • *Section 5.8.1* (*CE/UE Error Data Record Format*) • *Table 16: Boot Stage Registers*
1.11	January 31, 2020	Updated: • *Section 5.8* (*Core Error Register Definitions*) • CE/UE error format and registers (including Core, MCU, PCIe, and other registers); Added *Section 5.8.1* (*CE/UE Error Data Record Format*) • *Table 6: GPI Mask Register Definitions* • *Table 7: GPI Source Register Definitions* • *Table 9: GPI Status Register Definitions* • *Table 10: Core Register Definitions* • *Table 14: Memory Error Registers* • *Table 16: Boot Stage Registers* • *Table 18: PCIe Error Registers* (replaced) • *Table 19: Other Error Registers*
1.10	January 24, 2020	Updated: • *Table 5: Logical Power Sensor Register Definitions* • *Table 6: GPI Mask Register Definitions* • *Table 8: GPI Interrupt Alert Behaviors Definitions* • *Table 9: GPI Status Register Definitions* • *Table 10: Core Register Definitions* • *Table 14: Memory Error Registers* • *Table 16: Boot Stage Registers* • *Table 19: Other Error Registers* • *Section 5.14* (*Other Errors*)
1.00	December 15, 2019	Updated: • *Table 6: GPI Mask Register Definitions* • *Table 9: GPI Status Register Definitions* • *Table 10: Core Register Definitions* • *Table 14: Memory Error Registers* • *Table 15: RAS Internal Error Registers* • *Section 5.14* (*Other Errors*)
0.8	October 23, 2019	• Updated DIMM CE threshold and VRD fault GPI register descriptions. • Updated MCU registers. • Deleted Appendix A. • Minor updates and fixes.
0.7	April 24, 2019	Minor updates, fixes, and enhancements.

ISSUE	DATE	DESCRIPTION
0.6	April 01, 2019	Updated these tables: • Identification table • Core cluster register • Socket Info register • Default values • DIMM VRD fault register • L1/L2 register definition registers
0.5	March 31, 2019	• Updated System Level Cache registers. • Added boot log registers.
0.4	March 31, 2019	Added miscellaneous operations.
0.3	March 21, 2019	Minor updates and fixes.
0.2	February 05, 2019	Minor updates and fixes.
0.1	December 04, 2018	Initial issue.

February 9, 2023

Ampere Computing reserves the right to change or discontinue this product without notice.

While the information contained herein is believed to be accurate, such information is preliminary, and should not be relied upon for accuracy or completeness, and no representations or warranties of accuracy or completeness are made.

The information contained in this document is subject to change or withdrawal at any time without notice and is being provided on an "AS IS" basis without warranty or indemnity of any kind, whether express or implied, including without limitation, the implied warranties of non-infringement, merchantability, or fitness for a particular purpose.

Any products, services, or programs discussed in this document are sold or licensed under Ampere Computing's standard terms and conditions, copies of which may be obtained from your local Ampere Computing representative. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Ampere Computing or third parties.

Without limiting the generality of the foregoing, any performance data contained in this document was determined in a specific or controlled environment and not submitted to any formal Ampere Computing test. Therefore, the results obtained in other operating environments may vary significantly. Under no circumstances will Ampere Computing be liable for any damages whatsoever arising out of or resulting from any use of the document or the information contained herein.

Ampere Computing
4655 Great America Parkway, Santa Clara, CA 95054
Phone: (669) 770-3700
https://www.amperecomputing.com

Ampere Computing reserves the right to make changes to its products, its datasheets, or related documentation, without notice and warrants its products solely pursuant to its terms and conditions of sale, only to substantially comply with the latest available datasheet.

Ampere, Ampere Computing, the Ampere Computing and 'A' logos, Altra, and eMAG are registered trademarks of Ampere Computing.

Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All other trademarks are the property of their respective holders.